Optimal Cooperative Control of UAVs for Dynamic Data-Driven Monitoring Tasks

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Vom Fachbereich Informatik der
Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

Dissertation

von

Dipl.-Math. Juliane Euler geb. Kuhn
(geboren in Heppenheim a. d. Bergstraße)

Referent:	Prof. Dr. Oskar von Stryk
Korreferent:	Prof. Rui Rocha, Ph.D. (University of Coimbra, Portugal)
Tag der Einreichung:	13.12.2016
Tag der mündlichen Prüfung:	31.01.2017

D17
Darmstadt 2018

Abstract

In recent years, there has been an immense improvement of methods and technology for Unmanned Aerial Vehicles (UAVs). By now, a steadily growing number of affordable platforms, as well as accurate onboard sensors, are available, which enables the use of UAVs as a remote sensing tool in various monitoring, surveillance, and disaster response tasks.

This thesis deals with the challenging application scenario of monitoring atmospheric dispersion processes using multiple sensor-equipped UAVs. The idea is to enable the UAVs to autonomously move through the domain such that the utility of measurements taken along the way is maximized and cooperation among the team members is exploited.

A three-part solution to this problem is developed. For cooperative control of multiple vehicles, a decentralized model-predictive control approach is proposed that is based on a mixed-integer linear system description. As data-driven adaptive sensing strategy, a sequential optimum design approach for the computation of vehicle-specific sensing trajectories with maximized information value is presented. In the last step, both approaches are combined to form a decentralized dynamic data-driven cooperative feedback control scheme.

During development and implementation of all parts, particular attention is paid to efficiency and adaptability in order for the proposed scheme to be applied decentralized in possibly changing heterogeneous vehicle team constellations.

Though atmospheric dispersion monitoring by UAVs serves as the motivating use case throughout this thesis, the developed solution is not limited to this specific scenario. In fact, it can easily be modified to deal with other kinds of unmanned vehicles or dynamic processes, and can, therefore, be applied to many other related problem types.

Applicability, versatility, and effectiveness of the approach are successfully evaluated based on physics simulations of representative multi-objective monitoring scenarios.

Kurzdarstellung

Die Forschung an verbesserten Methoden und Technologien für unbemannte Flugsysteme (engl. Unmanned Aerial Vehicles (UAVs)) hat in den letzten Jahren einen immensen Aufschwung erfahren. Die Auswahl an erschwinglichen Plattformen sowie Onboard-Sensoren mit hoher Genauigkeit wächst stetig und ermöglicht die Nutzung von UAVs für die Fernerkundung in diversen Beobachtungs- und Überwachungsaufgaben sowie in der Katastrophenhilfe.

Diese Arbeit behandelt das anspruchsvolle Anwendungsszenario der Beobachtung atmosphärischer Ausbreitungsprozesse durch mehrere mit Sensoren ausgestattete UAVs. Hierbei soll ein Team autonomer UAVs sich derart durch ein Gebiet bewegen, dass die unterwegs gesammelten Messdaten von maximalem Nutzen sind und die Kooperation der Teammitglieder untereinander bestmöglich ausgenutzt wird.

Die hier vorgestellte Lösung dieses Problems besteht aus drei Teilen. Für die Regelung kooperativer Mehrfahrzeugsysteme wurde ein dezentraler modell-prädiktiver Regelungsansatz entwickelt, der auf gemischt-ganzzahligen linearen Systembeschreibungen basiert. Als adaptive datengesteuerte Messstrategie werden mittels sequentieller optimaler Versuchsplanung fahrzeugspezifische Messtrajektorien mit maximalem Informationsgehalt berechnet. In einem letzten Schritt werden beide Methoden dann zu einem neuartigen dezentralen und dynamisch datengesteuerten kooperativen Regelungsansatz kombiniert.

Während der Entwicklung und Implementierung aller Lösungskomponenten wurde besonderes Augenmerk auf numerische Effizienz und Anpassbarkeit gelegt, um die Anwendung des vorgestellten Ansatzes in dezentralen und sich verändernden heterogenen Fahrzeugkonstellationen zu ermöglichen.

Obwohl sich der Anwendungsfall der Beobachtung atmosphärischer Ausbreitungsprozesse durch UAVs als Motivation durch die gesamte Arbeit zieht, ist die Anwendbarkeit der entwickelten Lösung nicht auf dieses spezifische Szenario beschränkt. Sie kann leicht für die Verwendung anderer Fahrzeugtypen oder die Betrachtung anderer dynamischer Prozesse angepasst werden und eignet sich dadurch für viele verschiedene verwandte Fragestellungen.

Die Ergebnisse von durchgeführten Physik-Simulationen mehrerer repräsentativer multikriterieller Beobachtungsszenarien demonstrieren die Anwendbarkeit, Vielseitigkeit und Effektivität der entwickelten Methoden.

Contents

List of Figures

List of Acronyms

ADE	Advection Diffusion Equation
CFTOC	Constrained Finite Time Optimal Control
CMOMMT	Cooperative Multi-Robot Observation of Multiple Moving Targets
DDDAS	Dynamic Data-Driven Application System
DPS	Distributed Parameter System
ETKF	Ensemble Transform Kalman Filter
HOCP	Hybrid Optimal Control Problem
iff	if and only if
LP	Linear Programming
MILP	Mixed Integer Linear Program
MIP	Mixed Integer Program
MIQP	Mixed Integer Quadratic Program
MLD	Mixed Logical Dynamical
MPC	Model-Predictive Control
NLP	Nonlinear Program
OCP	Optimal Control Problem
ODE	Ordinary Differential Equation
OED	Optimum Experimental Design
PDE	Partial Differential Equation
PWA	Piecewise Affine
RHC	Receding Horizon Control
RMSE	Root Mean Squared Error
ROS	Robot Operating System
UAV	Unmanned Aerial Vehicle
URDF	Unified Robot Description Format
wNLLSQ	weighted Nonlinear Least Squares

List of Symbols

List of important recurring mathematical symbols and notations in the order of their occurrence.

$\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$	bold face lowercase letters denote vectors or vector-valued functions
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots$	bold face uppercase letters denote matrices
$\mathbf{x}$	state vector $\in \mathbb{R}^{n_x}$
$\mathbf{u}$	vector of control inputs $\in \mathbb{R}^{n_u}$
$\boldsymbol{\delta}$	vector $\in \{0,1\}^{n_\delta}$ of auxiliary binary variables in MLD systems
$\mathbf{z}$	vector $\in \mathbb{R}^{n_z}$ of auxiliary continuous variables in MLD systems
$\mathbf{x}(t)$	vector $\mathbf{x}$ at time $t \in \mathbb{R}$
$\mathbf{x}^k := \mathbf{x}(t_k)$	vector $\mathbf{x}$ at the discrete time step $t_k, k \in \mathbb{N}$
$\Delta t := t_{k+1} - t_k$	step size in discrete-time systems
T	overall number of considered time steps
N	length of MPC prediction horizon
$\forall$	for all
$\exists$	there exists
$\equiv$	logical equivalence
$\mathbf{A} \succ 0$	matrix $\mathbf{A}$ is positive definite
$\mathbf{A} \succeq 0$	matrix $\mathbf{A}$ is positive semidefinite
n_d	number of hyperplanes for linear approximation of Euclidean distances
n_R	overall number of robots in the system
n_T	overall number of robots in the targets
$\widetilde{n}_R$	number of robots in a local subsystem
$\widetilde{n}_T$	number of targets in a local subsystem
$\widetilde{n}_{Rmax}$	maximum number of robots in a local subsystem
$\widetilde{n}_{Tmax}$	maximum number of targets in a local subsystem
n_V	overall number of vehicles in the system
$\widetilde{n}_V$	number of vehicles in a local subsystem
d_{com}	communication range
d_{col}	minimum safety distance for collision avoidance
$v_{\max}$	maximum velocity for vehicles modeled as point mass
$u_{\max}$	maximum acceleration for vehicles modeled as point mass

$v_{z\max}$	maximum climb/descent rate for airplane model
$u_{z\max}$	maximum z acceleration for airplane model
φ	orientation of airplane model
$\omega_{\varphi\max}$	maximum angular velocity for airplane model
$u_{\varphi\max}$	maximum angular acceleration for airplane model
s	constant forward speed of airplane model
$c(\mathbf{x}, t)$	contaminant concentration at $\mathbf{x} = (x, y, z)$ and time t in [kg/m^3]
$\mathbf{v}_w = (v_{wx}, v_{wy}, v_{wz})$	wind velocity with x, y, and z component in [m/s]
K_x, K_y, K_z	turbulent x, y and z eddy diffusivities in [m^2/s]
Q	contaminant mass in [kg]
(x_0, y_0, z_0)	source location of contaminant release in [m]
t_0	time of instantaneous contaminant release in [s]
$\boldsymbol{\theta}$	vector of parameters $(Q, K_x, x_0, y_0, z_0, t_0)$ of the Gaussian puff model
$\boldsymbol{\theta}_{\text{true}}$	true parameter vector
$\bar{\boldsymbol{\theta}}$	current parameter estimate
$\bar{\boldsymbol{\theta}}_0$	initial parameter estimate
$C(\boldsymbol{\theta}, x, y, z, t)$	contaminant concentration at (x, y, z, t) according to the Gaussian puff model parameterized by $\boldsymbol{\theta}$ in [kg/m^3]
$(\mathbf{p}, v)$	data point consisting of location $\mathbf{p} = (x, y, z, t)$ and value v
ε	measurement error
σ	standard deviation of the measurement error
$\mathbf{W}$	diagonal weight matrix
$\lvert\mathbf{A}\rvert$	determinant of matrix $\mathbf{A}$
ξ	continuous design
ξ_m	exact design with m trials
Ξ	design region
$\mathbf{M}(\xi)$	information matrix for continuous design ξ
$\mathbf{M}(\xi_m)$	information matrix for exact design ξ_m
$\text{tr}(\mathbf{A})$	trace of matrix $\mathbf{A}$
∇	gradient
$\mathbf{I}$	identity matrix
Δt_m	time between consecutive measurements in s

1 Introduction

Research on Unmanned Aerial Vehicles (UAVs) and their applications has increased tremendously in the last decade with a clear focus on aerial imagery and remote sensing [102, 108, 36]. While images or sensor data captured from above can be useful for all kinds of environmental, agricultural, or urban surveillance and monitoring tasks, they can be especially valuable for post-disaster assessment, e.g. after an earthquake or hurricane, even more, if the infrastructure is severely damaged and the area of interest is not accessible from the ground.

The articles [50] and [10] provide surveys of aerial imaging applications in the context of disaster management. An experimental evaluation of emergency response applications for UAVs is presented in [17]. Hazards that can exclusively or additionally be assessed by non-visual remotely sensed data include the atmospheric dispersion of volcanic ash [110], as experienced in 2011 with the Icelandic Eyjafjallajökull Volcano, or other kinds of airborne material. Radiation surveillance by UAVs [111] is another possible application in this context, represented by the prominent example of the Fukushima nuclear disaster in 2011.

The Icelandic ash hazard and the Fukushima nuclear accident are examples of rare major disasters with far-reaching consequences. However, incidents causing atmospheric dispersion of hazardous material at a smaller scale are not uncommon as shown by some recent examples depicted in Figure 1.1.
In these scenarios, UAVs can be extremely helpful for the collection of sensor data. In contrast to stationary sensor nodes, sensor-carrying UAVs can be deployed flexibly and move to the most relevant measurement locations. That way, no humans are required to enter a hostile or possibly life-threatening environment and the rescuers can focus on the interpretation of information rather than its acquisition.

However, as sensors gain mobility when mounted on a UAV (or other types of unmanned vehicles), the immediate question arises where to send the vehicles in order to obtain the most relevant information. As stated in [16], human experts in atmospheric dispersion and humans able to teleoperate aerial systems might not be on hand or pose a significant cost factor for the mission, especially as, in comparison, the price for robotic systems offering a certain level of autonomy keeps falling. Moreover, the range of the onboard communication devices might not suffice to guarantee a permanent link to an operator station. Autonomous behavior can close these gaps by enabling the UAVs to take decisions and plan their movements on their own. Depending on the complexity of the situation and on the algorithms defining a UAV's decision logic, autonomously selected actions can even outperform those commanded by an operator since relevant facts might not be obvious or intuitive to humans.

On the growing UAV market, more and more hardware becomes available including light-weight accurate sensors for a variety of sensing tasks. With the number of affordable platforms increases the interest in using multiple instead of a single UAV to perform a task. Involving more than one UAV promises a faster task completion and higher quality of the obtained solution, making the mission significantly more efficient. At the same time, however, the question arises how

(a) Barcelona, ESP, 2015: Due to an explosion in a chemical plant, toxic and caustic material including nitric acid is released. (Source: [8])

(b) Los Angeles, US, 2015: For more than 4 months, tons of methane and ethane can escape through a leak in a well within an underground storage facility. (Source: [7])

(c) Hamburg, GER, 2015: A mixture of steam and toxic lye escapes through a leak in a tank of a chemical plant. (Source: [5])

(d) Litvinov, CZE, 2015: An explosion in a chemical plant causes a nontoxic black cloud of smoke to drift towards Germany. (Source: [6])

Figure 1.1: Examples of recent incidents causing atmospheric dispersion of hazardous material.

coordination or even cooperation of multiple autonomous vehicles can be achieved and how it can improve the overall task performance.

In this context, the problem considered in this thesis can be stated as follows:

How can the trajectories of a team of (possibly heterogeneous) sensor-equipped autonomous vehicles be controlled in an optimal manner, such that the estimation of an atmospheric dispersion process is continuously improved by assimilating the measurement data obtained from the onboard sensors?

1.1 Problem Characteristics and Challenges

Atmospheric dispersion is a highly-dynamic spatiotemporal process, the identification of which can hardly be accomplished in real-time by applying computationally intensive feedforward path planning for the sensor carrying vehicles. Instead, adaptation to the permanently changing environment and efficiency, ideally real-time efficiency, of the employed algorithms are desired key features of a suitable control approach. In addition, as the contaminant plume can take

on vast spatial proportions, complete sensor coverage might be impossible and an adaptive movement strategy is required for guiding the sensor vehicles to the currently most informative measurement locations.

The dimensions of the domain in which the vehicles operate also make it difficult to maintain a permanent communication link from each vehicle to a central control instance, but also from one vehicle to another. Therefore, a decentralized system and control architecture better suits the considered application scenario. It is flexible and scalable in the number of involved team members and therefore more robust to vehicle or communication failures than centralized approaches.

In order to best exploit a vehicle's potential as sensor platform, the control approach should allow for its individual physical motion capabilities and combine the skills of different team members in a way that is most beneficial for the global mission objective. This can be best achieved by employing optimization methods based on motion dynamics models of the involved vehicles and on mathematical formulations of how their cooperation affects the team's common goal.

In essence, the considered problem comprises two aspects: multi-vehicle cooperation and spatiotemporal process estimation. The challenge is to meld approaches from both worlds to obtain a new method that meets the requirements derived above:

- **efficiency** enabling real-time application,
- **adaptability** to the changing environment and the collected data,
- **decentrality** of all components,
- **scalability** to a varying number of vehicles,
- **flexibility** to handle changing team configuration, and
- **optimality** in path planning, cooperation, and estimation.

It has to be noted, that some of these aspects are interdependent and that some of them can only be achieved at the expense of others. The numerical efficiency required for an optimization-based decentralized feedback control approach applicable in real-time on board a UAV with limited computation resources can only be achieved by using approximations to reduce the complexity of the underlying mathematical models of both the vehicle dynamics as well as the dispersion process. Finding an adequate trade-off between accuracy and efficiency is another challenge in this context.

1.2 Scientific Contributions and Outline

This thesis aims at solving the challenging coupled cooperative control and estimation problem motivated above in a highly efficient, yet optimization-based manner that meets all requirements derived in the previous section.

To the field of **cooperative control**, this thesis contributes a decentralized model-predictive feedback controller that is based on discrete-continuous linear models of the cooperative multi-vehicle system. It is versatilely applicable to many kinds of cooperative mobility problems and its implementation allows flexible online adaptation to changing team configurations.

To the field of **adaptive mobile sensing**, this thesis contributes a method for optimizing the trajectories of sensor-equipped vehicles with respect to maximum improvement in estimating the parameters of a dynamic process model. The approach employs sequential optimum design and respects vehicle-specific motion dynamics constraints.

To the field of **Dynamic Data-Driven Application Systems** (DDDAS), this thesis contributes a decentralized data-driven feedback control scheme by coupling the developed cooperative controller with the developed sequential design for adaptive sensing. The proposed approach enhances the concept of DDDAS by the ability to simultaneously solve the sensing task as well as additional cooperative mobility tasks.

In the recent survey [44], the authors identified, among others, adaptive path planning for dynamic process tracking (in real-time), the cooperation of heterogeneous sensor platforms, and decentralized cooperative control (with limited communication) as open research challenges in the field of robotics for environmental monitoring. This emphasizes the relevance of the proposed solutions that will be detailed in the following chapters, as outlined below.

Chapter 2

After a brief review of the state of the art in cooperative control, the theoretical background on mixed-integer Model-Predictive Control (MPC), i.e. MPC of discrete-continuous linear systems, is summarized. Section 2.4 describes how the concept of mixed-integer MPC has been adopted in this thesis for developing an efficient decentralized feedback control approach for cooperative multi-vehicle systems. A simulation-based validation of the controller in different application scenarios is provided. Finally, short descriptions of different application scenarios investigated for a simulation-based validation of the model-predictive controller along with selected modeling details and results are given at the end of the chapter.

Chapter 3

Chapter 3 first summarizes related work in adaptive sensing with mobile sensor platforms followed by a survey of the theoretical background in modeling atmospheric dispersion, parameter estimation, and sequential optimum design as far as it is relevant for this thesis. Section 3.4 describes the calculation of vehicle-specific waypoints based on sequential optimum design. The obtained waypoints define optimized sensor vehicle trajectories such that measurements gathered during movement maximize the quality of the parameter estimation of a Gaussian puff model of an atmospheric dispersion process. By simulation-based evaluation, different aspects of the approach and their effect on its performance are illustrated. The results are presented in Section 3.5.

Chapter 4

This chapter treats the combination of the developments from Chapters 2 and 3 to obtain a novel dynamic data-driven cooperative sensing and control approach. After a brief overview of related work on dynamic process monitoring by cooperating sensor-equipped unmanned vehicles, the developed coupling of mixed-integer MPC and sequential optimum design and selected implementation details of the employed realistic simulation framework are presented. The concept is evaluated in terms of a simulated multi-objective monitoring scenario described in Section 4.5. Section 4.6 outlines further potential applications the proposed scheme is able to deal with.

Chapter 5

The thesis concludes with a summary and evaluation of the contributions detailed in Chapters 2 to 4 and an outlook on interesting research directions to follow up.

2 Decentralized Model-Predictive Control of Cooperative Multi-Vehicle Systems

In this chapter, an overview of existing approaches for cooperative control of multi-vehicle systems with focus on optimization-based methods is given in Section 2.1. Section 2.2 then states how the decentralized Model-Predictive Control (MPC) scheme proposed in this thesis contributes to this field.

The theoretical background on hybrid dynamical systems, their abstraction by discretization and linearization, as well as the concepts and characteristics of mixed-integer MPC, are summarized in Section 2.3 as it provides the basis for the proposed control scheme. How the theoretical concepts are adopted to derive a decentralized cooperative model-predictive controller is described in Section 2.4 followed by a simulation-based analysis and validation by means of different application scenarios presented in Section 2.5.

2.1 Related Work in Cooperative Control

"Cooperative control has become one of the most popular control topics in the last decade." [19]

It is therefore hardly possible to overview the entire field of research in multi-robot task allocation and existing cooperative control approaches, although several reviews and survey papers exist, e.g. [93, 74, 151, 90, 9].

The multi-vehicle tasks considered in this thesis are characterized by a tight coupling of discrete decisions (e.g. target assignment) and optimal continuous trajectory planning. The vehicles' motion capabilities have a significant influence on the task performance. This problem type is therefore referred to as *cooperative mobility problem* as introduced in [119]. Examples include target observation, patrol, intrusion detection, exploration, coverage, or formations by multiple cooperating autonomous vehicles. Neither solving the pure task assignment problem – solely considered an NP-hard problem already – nor motion planning alone will be sufficient to achieve optimal cooperative mobility. The same holds for a subsequent consideration of these two problem aspects, e.g. as in [99]. Role assignment plus the robots' motion dynamics have to be taken into account simultaneously, further increasing the problem complexity.

Various heuristic approaches have been proposed that rely on different assignment protocols, i.e. a set of decision rules that determine the task assigned to each robot [93], such as behavior-based (e.g. [104, 86, 43, 89, 124, 141]) or market-based methods (e.g. [92, 72], reviewed in [42]). Other possible ways to achieve cooperative behavior of autonomous vehicles include multi-robot learning (e.g. [107, 103]), probabilistic (e.g. [132, 114, 115]), game-theoretic [112], and graph-based strategies (e.g. [113, 75]), or even "hybrid heuristics", such as the combination of Greedy search and Simulated Annealing proposed in [63]. However, these approaches do not guarantee optimality.

Optimization-Based Cooperative Feedforward Control

In optimization-based cooperative control methods, (control) trajectories are determined based on a model of the multi-vehicle system consisting of a cost function representing the mission objective(s) subject to constraints describing the vehicles' motion dynamics and their intended cooperation. Modeling cooperative mobility, therefore, results in a hybrid dynamical system comprising continuous as well as discrete variables, differential equations, and logical rules. Optimization based on such a system leads to a (possibly nonlinear) Hybrid Optimal Control Problem (HOCP) [23, 53, 119].

Despite ongoing research on the development and benchmarking of efficient numerical HOCP solvers (e.g. [129]), solving such problems is still computationally intractable, even for offline feedforward applications. A model approximation by time-discretization and linearization leads to the class of Mixed Integer Linear Programs (MILPs), for which highly efficient solvers exist. Therefore, many researchers employed this technique for developing multi-vehicle feedforward control approaches, e.g. for multi-robot games [45, 29, 120].

Model-Predictive Cooperative Control

A feedback control solution, however, is much more desirable for cooperative mobility problems. In contrast to feedforward control, by feedback, deviations of the true vehicle motion from their modeled behavior can be compensated and the cooperative strategy can be adapted in reaction to a dynamically changing environment.

An extensively used way to obtain closed-loop optimal cooperative control is to employ MPC based on MILPs or Mixed Logical Dynamical (MLD) systems, offering additional benefits like robustness and stability [15]. Instead of an a priori computation of vehicle trajectories over a long time horizon, in MPC, the MILP is repeatedly solved over a short time horizon (prediction horizon) only, further increasing the controller's computational efficiency. In this context, MPC can be seen as (horizontal) *time decomposition* of the cooperative control problem as opposed to (vertical) *functional decompositions*, such as a hierarchical framework with high-level layers for task assignment and path planning and low-level motion control [28, 84]. Centralized MILP-based MPC for multi-vehicle cooperation has, e.g., been applied in [83, 101].

Decentralized Model-Predictive Control

Depending on the size of the team, MPC performed centrally by a single controller deciding on the actions of all vehicles might still be too time-consuming, such that the optimization problem cannot be solved within the sampling period of the system to be controlled. Further drawbacks of centralized MPC are the dependency on a central processing unit, i.e. a central point of failure, and a stable communication to it. Centralized concepts are therefore not suitable for flexible system structures, such as plug-and-play systems [94].

That is why distributed and decentralized MPC approaches have been increasingly pursued in the last decade. In this context, [94] provides a comprehensive overview and classification of existing approaches according to 38 different features, and states about the advantages of distributed/decentralized MPC: "*The potential of distributed MPC lies in the unique combination of the strengths of MPC (namely, feedback with feed-forward control in a receding horizon fashion, multi-objective optimization, and explicitly handling of constraints) with the negotiation and coordination possibilities provided by communication.*"

Note, that definitions of decentralized versus distributed control are not consistent in the literature. Within the scope of this thesis, the term “decentralized” will generally refer to systems without a central decision unit. No single node has complete system knowledge but is able to exchange information with nearby teammates. Based on its local system information, each node individually determines its next action.

MILP-based MPC is decentralized by breaking down the overall system into smaller subproblems running their individual MPC loop. Subsystems can be defined in various ways, e.g. hierarchically [20], sequentially [122], graph-based [148], or based on a communication topology [51]. Also, the tightness of the coupling between two subsystems and the type of information they share differ in existing approaches.

2.2 Contribution

The approach for cooperative multi-vehicle control proposed in this thesis adopts the idea of hybrid optimal control with increased computational efficiency by abstraction, i.e. time discretization and linearization, of the system model. It is particularly suitable for solving cooperative mobility problems as it integrates task and motion planning by employing discrete-continuous linear system models. In the context of cooperative mobility in static environments, this technique has similarly been investigated for centralized feedforward control in [119]. Here, it is extended by employing MPC in a decentralized manner in order to obtain real-time efficient feedback control applicable in dynamically changing conditions.

In the review [94], the control of hybrid systems has been identified as a still underrepresented branch of research in the field of decentralized MPC. In addition, further research was suggested on control schemes that are able to handle dynamically changing system sizes.

Both gaps can be bridged by the proposed decentralized model-predictive controller. Optimized control inputs are determined based on each vehicle’s local subsystem state that includes information on teammates within communication range, but is independent of the overall system setup. That way, each vehicle’s individual controller is robust to changes in the overall number of involved vehicles and its implementation allows flexible online adaptation to changing heterogeneous team configurations. The trade-off between efficiency and accuracy of the approach can be scaled by tuning different parameters, e.g. subsystem size, approximation accuracy, or length of the prediction horizon. By modifying the system model and mission objective appropriately, the MPC approach is versatilely applicable to many kinds of cooperative mobility problems, which is confirmed by simulation results for several different multi-vehicle scenarios.

2.3 Background

2.3.1 Characteristics of Hybrid Dynamical Systems

There exist many different modeling frameworks for hybrid systems, each with their own field of application, an overview can be found in [40]. What they all have in common and what

represents the essential feature of hybrid dynamical systems, is the combination of different dynamics. Typically, time-driven model parts (differential or difference equations) interact with event-driven model parts (e.g. finite-state machines, automata) or logical rules.

One example of a modeling formalism is given in the following definition that extends the mathematical concept of differential inclusions to *hybrid inclusions*. This results in a very compact and clearly structured model description that is, yet, able to describe a broad range of hybrid phenomena.

Definition 2.3.1. ***Hybrid system.*** A general model $\mathcal{H}$ of a hybrid system is given by

$$\dot{\mathbf{x}} \in \mathbf{F}(\mathbf{x}), \quad \text{if } \mathbf{x} \in \mathcal{C}\,, \tag{2.1a}$$

$$\mathbf{x}^{+} \in \mathbf{G}(\mathbf{x}), \quad \text{if } \mathbf{x} \in \mathcal{D}\,, \tag{2.1b}$$

where $\mathbf{F}$ is a set-valued map called the *flow map* representing continuous-time dynamics. The differential inclusion (2.1a) is constrained by the *flow set* $\mathcal{C} \subset \mathbb{R}^n$. Analogously, $\mathbf{G}$ is a set-valued map called the *jump map* representing discrete-time dynamics and the difference inclusion (2.1b) is constrained by the *jump set* $\mathcal{D} \subset \mathbb{R}^n$. It is required that $\mathbf{F}(\mathcal{C}) \neq \emptyset$ and $\mathbf{G}(\mathcal{D}) \neq \emptyset$.

A variety of alternative hybrid system descriptions, including hybrid automata and switched systems, can be translated to the general model (2.1). Hybrid inclusions are useful for studying general properties of hybrid systems, such as stability and robustness. For more details, the reader is referred to [54] and the recent book [55] by the same authors.

2.3.2 Modeling Discrete-Time Linear Hybrid Dynamical Systems

For modeling the cooperative mobility tasks considered in this thesis, the employed hybrid modeling framework needs to capture the interaction of continuous motion dynamics represented by ordinary differential equations and logical rules, the formulation of which contains binary state variables.

Since, in addition, emphasis is put on high computational efficiency required to apply the control scheme online with limited onboard computing capacities, time-discrete linear model formulations are used.

A well-established and suitable representation of time-discrete linear hybrid system is provided by the MLD Framework that will be introduced in the following. It offers particular benefits within an MPC scheme as will be discussed in Section 2.3.4.

Mixed-Logical Dynamical Systems

The MLD framework was first proposed in [15] for modeling and controlling constrained linear discrete-time systems containing interacting physical laws and logical rules. In this thesis, time-invariant system matrices are considered, which are derived from a suitable abstraction and discretization of the full mathematical problem formulation.

A system with linear dynamic equations subject to mixed-integer inequalities is obtained:

$$\mathbf{x}^{k+1} = \mathbf{A}\mathbf{x}^k + \mathbf{B}_1\mathbf{u}^k + \mathbf{B}_2\boldsymbol{\delta}^k + \mathbf{B}_3\mathbf{z}^k \tag{2.2a}$$

$$\mathbf{y}^k = \mathbf{C}\mathbf{x}^k + \mathbf{D}_1\mathbf{u}^k + \mathbf{D}_2\boldsymbol{\delta}^k + \mathbf{D}_3\mathbf{z}^k \tag{2.2b}$$

$$\mathbf{E}_2\boldsymbol{\delta}^k + \mathbf{E}_3\mathbf{z}^k \leq \mathbf{E}_1\mathbf{u}^k + \mathbf{E}_4\mathbf{x}^k + \mathbf{E}_5\ , \tag{2.2c}$$

where $k \in \mathbb{Z}$ is the current time step,
$\mathbf{x} = (\mathbf{x}_c^T, \mathbf{x}_b^T)^T$ with $\mathbf{x}_c \in \mathbb{R}^{n_x^c}, \mathbf{x}_b \in \{0,1\}^{n_x^b}, n_x = n_x^c + n_x^b$ describes the system state,
$\mathbf{y} = (\mathbf{y}_c^T, \mathbf{y}_b^T)^T$ with $\mathbf{y}_c \in \mathbb{R}^{n_y^c}, \mathbf{y}_b \in \{0,1\}^{n_y^b}, n_y = n_y^c + n_y^b$ is the output vector,
$\mathbf{u} = (\mathbf{u}_c^T, \mathbf{u}_b^T)^T$ with $\mathbf{u}_c \in \mathbb{R}^{n_u^c}, \mathbf{u}_b \in \{0,1\}^{n_u^b}, n_u = n_u^c + n_u^b$ is the control input,
and $\boldsymbol{\delta} \in \{0,1\}^{n_\delta}$ and $\mathbf{z} \in \mathbb{R}^{n_z}$ represent auxiliary binary and continuous vectors, respectively.

In general, the inequalities (2.2c) can hold for different values of $\mathbf{z}$ and $\boldsymbol{\delta}$. However, a unique dependence of $\mathbf{x}^{k+1}$ and $\mathbf{y}^k$ on $\mathbf{x}^k$ and $\mathbf{u}^k$ is desirable, which motivates the following definition of *well-posedness*. In the context of differential equation systems, the term typically means that for any initial condition solutions exist and are unique within a specified function class. In that case, the model defines a mapping from the set of initial conditions to the set of trajectories. Note, however, that in contrast to smooth dynamical systems, for hybrid systems this mapping might not be continuous.

Definition 2.3.2. ***Well-posedness [15].*** Let $\mathcal{I}_\mathbf{B}$ denote the set of all indices $i \in \{1, \ldots, n_\delta\}$, such that $[\mathbf{B}_2]^i \neq 0$, where $[\mathbf{B}_2]^i$ denotes the ith column of $\mathbf{B}_2$. Let $\mathcal{I}_\mathbf{D}$, $\mathcal{J}_\mathbf{B}$, $\mathcal{J}_\mathbf{D}$ be defined analogously by collecting the positions of nonzero columns of $\mathbf{D}_2$, $\mathbf{B}_3$, and $\mathbf{D}_3$, respectively. Let $\mathcal{I} \triangleq \mathcal{I}_\mathbf{B} \cup \mathcal{I}_\mathbf{D}$ and $\mathcal{J} \triangleq \mathcal{J}_\mathbf{B} \cup \mathcal{J}_\mathbf{D}$. An MLD system (2.2) is said to be *well-posed* if, $\forall k \in \mathbb{Z}$,

(i) $\mathbf{x}^k$ and $\mathbf{u}^k$ satisfy (2.2c) for some $\boldsymbol{\delta}^k \in \{0,1\}^{n_\delta}$, $\mathbf{z}^k \in \mathbb{R}^{n_z}$, and $\mathbf{x}_b^{k+1} \in \{0,1\}^{n_x^b}$;

(ii) $\forall i \in \mathcal{I}$ there exists a mapping $\mathcal{D}_i : \mathbb{R}^{n_x+n_u} \to \{0,1\}$ such that the ith component $\delta_i^k = \mathcal{D}_i(\mathbf{x}^k, \mathbf{u}^k)$, and $\forall j \in \mathcal{J}$ there exists a mapping $\mathcal{Z}_j : \mathbb{R}^{n_x+n_u} \to \mathbb{R}$ such that $z_j^k = \mathcal{Z}_j(\mathbf{x}^k, \mathbf{u}^k)$.

The functions $\mathcal{D}_i$ and $\mathcal{Z}_j$ are implicitly defined by (2.2c) and they are nonlinear due to the integer constraint $\delta_i \in \{0,1\}$.

Remark 2.3.1. [14]. An MLD system is said to be *completely well-posed* if in addition $\mathcal{I} = \{1, \ldots, n_\delta\}$ and $\mathcal{J} = \{1, \ldots, n_z\}$. In other words, this means that for all $\mathbf{x}$ and $\mathbf{u}$ the auxiliary variables $\boldsymbol{\delta}$ and $\mathbf{z}$ are uniquely determined and therefore $\mathbf{x}^{k+1}$ and $\mathbf{y}^k$ are uniquely defined by $\mathbf{x}^k$ and $\mathbf{u}^k$.

The term well-posedness cannot only be applied to MLD systems, but also to single auxiliary variables.

Definition 2.3.3. ***Well-posed variables [15].*** An auxiliary variable δ_i^k (z_i^k) is *well-posed* if $i \in \mathcal{I}$ ($j \in \mathcal{J}$) and *indefinite* otherwise.

MLD model descriptions are equivalent to discrete-time Piecewise Affine (PWA) systems described as

$$\left.\begin{array}{rcl} \mathbf{x}^{k+1} & = & \mathbf{A}_i\mathbf{x}^k + \mathbf{B}_i\mathbf{u}^k + \mathbf{a}_i \\ \mathbf{y}^k & = & \mathbf{C}_i\mathbf{x}^k + \mathbf{D}_i\mathbf{u}^k + \mathbf{c}_i \end{array}\right\} \quad \text{if} \begin{pmatrix} \mathbf{x}^k \\ \mathbf{u}^k \end{pmatrix} \in \mathcal{P}_i\ , \tag{2.3}$$

where $i = 1, 2, \ldots, n$ represent discrete operation modes and $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$ are convex non-overlapping polyhedra [14]. A polyhedron $\mathcal{P}_i$ can be described by inequality constraints on $\mathbf{x}$ and $\mathbf{u}$, so called *guard lines*

$$\mathbf{G}_i^x \mathbf{x}^k + \mathbf{G}_i^u \mathbf{u}^k \leq \mathbf{G}_i^c. \tag{2.4}$$

All theoretical findings for MLD systems also hold for PWA systems and vice versa.

In the remainder of this subsection, the necessary steps for transforming continuous-time nonlinear dynamics and mixed-logical constraints into discrete-time linear expressions compatible with the MLD framework (2.2) are described.

Hybridization

Hybridization means the approximation of nonlinear model parts by piecewise affine functions. The technique employed in this thesis was first proposed in [119].

Let

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{2.5}$$

be a differential equation with nonlinear right hand side $\mathbf{f}$ defined on $\mathbb{X} \times \mathbb{U} \subset \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$ and let the states $\mathbf{x}$ and controls $\mathbf{u}$ satisfy the equality and inequality constraints

$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) = 0 \quad \text{and} \tag{2.6a}$$

$$\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \leq 0\,. \tag{2.6b}$$

The idea is to cover $\mathbb{X} \times \mathbb{U}$ by a finite set of simplices $\mathbb{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n\}$ such that for every two elements of $\mathbb{S}$: $(\mathcal{S}_i \cap \mathcal{S}_k) \subset (\partial\mathcal{S}_i \cap \partial\mathcal{S}_k)$ for $i \neq k$. On each simplex, the original function $\mathbf{f}$ is approximated by an affine function. Hence, a piecewise affine system description of the form

$$\dot{\mathbf{x}}(t) = \mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t) + \mathbf{a}_i \quad \text{if} \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix} \in \mathcal{S}_i \tag{2.7}$$

with $i = 1, 2, \ldots, n$, time-invariant matrices $\mathbf{A}_i$, $\mathbf{B}_i$, and time-invariant vector $\mathbf{a}_i$ is obtained.

The vertices $(\mathbf{x}_j^i, \mathbf{u}_j^i)$, $j = 1, 2, \ldots, n_i$ of a simplex $\mathcal{S}_i$ and corresponding values $\mathbf{z}_j^i$ are determined by solving the nonlinear optimization problem

$$\min_{\{\mathbf{x}_j^i, \mathbf{u}_j^i, \mathbf{z}_j^i\}} \quad \sum_{i=1}^{n} \int_{\mathcal{S}_i} \|\mathbf{f}(\boldsymbol{\xi}, \boldsymbol{\nu}) - (\mathbf{A}_i \boldsymbol{\xi} + \mathbf{B}_i \boldsymbol{\nu} + \mathbf{a}_i)\|_2^2 \, d(\boldsymbol{\xi}, \boldsymbol{\nu}) \tag{2.8a}$$

$$\text{s.t.} \quad \mathbf{z}_j^i = \mathbf{A}_i \mathbf{x}_j^i + \mathbf{B}_i \mathbf{u}_j^i + \mathbf{a}_i \tag{2.8b}$$

$$\mathbf{g}(\mathbf{x}_j^i, \mathbf{u}_j^i) = 0 \tag{2.8c}$$

$$\mathbf{h}(\mathbf{x}_j^i, \mathbf{u}_j^i) \leq 0 \tag{2.8d}$$

$$\mathbb{X} \times \mathbb{U} \subseteq \bigcup \mathbb{S}\,, \tag{2.8e}$$

where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, n_i$, such that the overall deviation of the piecewise affine system from the original function is minimized. The number of simplices n has to be chosen in consideration of the desired accuracy of the PWA system versus the model complexity and the resulting computational effort.

Time discretization means the transition from a continuous time domain to discrete points in time t_k, $k = 1, 2, \ldots$, such that

$$t_{k+1} = t_k + \Delta t \ .$$

Δt is called the *step size* and may be constant or change adaptively over time. Instead of solving an equation for continuous time, the idea is to approximate the solution at each of the discrete time steps t_k, such that the approximate solution converges to the exact solution for $\Delta t \to 0$.

Euler's Method

The easiest way to obtain a difference equation by discretizing a differential equation (2.5) is provided by the forward Euler method.

Since an approximation of the derivative of $\mathbf{x}^k := \mathbf{x}(t_k)$ is given by

$$\frac{\mathbf{x}^{k+1} - \mathbf{x}^k}{\Delta t} \approx \dot{\mathbf{x}}^k = \mathbf{f}(\mathbf{x}^k, \mathbf{u}^k) \ , \tag{2.9}$$

it is straightforward to derive the explicit difference equation

$$\mathbf{x}^{k+1} \approx \mathbf{x}^k + \Delta t \cdot \mathbf{f}(\mathbf{x}^k, \mathbf{u}^k) \ . \tag{2.10}$$

The global discretization error of the Euler method is $\mathcal{O}(\Delta t)$, i.e. it provides a very rough approximation only and converges relatively slowly to the exact solution. However, it is easy to implement and requires only a single function evaluation per step. Meanwhile, it is still reasonably accurate when used over short time horizons.

Exact Discretization of Piecewise Affine Systems

If the system dynamics are given as continuous-time (piecewise) affine state space model of the form

$$\left.\begin{array}{rcl} \dot{\mathbf{x}}(t) & = & \mathbf{A}_i^C \mathbf{x}(t) + \mathbf{B}_i^C \mathbf{u}(t) + \mathbf{a}_i^C \\ \mathbf{y}(t) & = & \mathbf{C}_i^C \mathbf{x}(t) \end{array}\right\} \quad \text{if } \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix} \in \mathcal{S}_i \tag{2.11}$$

with $i = 1, 2, \ldots, n$, time-invariant matrices $\mathbf{A}_i^C$, $\mathbf{B}_i^C$, $\mathbf{C}_i^C$, and time-invariant vector $\mathbf{a}_i^C$, then according to [64] (with references to [130] and [87]) a discrete-time system representation

$$\left.\begin{array}{rcl} \mathbf{x}^{k+1} & = & \mathbf{A}_i^D \mathbf{x}^k + \mathbf{B}_i^D \mathbf{u}^k + \mathbf{a}_i^D \\ \mathbf{y}^k & = & \mathbf{C}_i^D \mathbf{x}^k \end{array}\right\} \quad \text{if } \begin{pmatrix} \mathbf{x}^k \\ \mathbf{u}^k \end{pmatrix} \in \mathcal{S}_i \tag{2.12}$$

for time steps k and constant step size $\Delta t := t_{k+1} - t_k$ can be determined via the formulas

$$\mathbf{A}_i^D = e^{\mathbf{A}_i^C \Delta t}, \quad \mathbf{B}_i^D = \int_0^{\Delta t} e^{\mathbf{A}_i^C \nu} d\nu \cdot \mathbf{B}_i^C, \quad \mathbf{C}_i^D = \mathbf{C}_i^C, \quad \text{and} \quad \mathbf{a}_i^D = \int_0^{\Delta t} e^{\mathbf{A}_i^C \nu} d\nu \cdot \mathbf{a}_i^C \ .$$

This holds under the assumption that the control inputs are constant between time steps, i.e. $\mathbf{u}(t) = \mathbf{u}^k = \text{const}$ for $t \in [t_k, t_{k+1})$. The obtained discrete values $\mathbf{x}^k$ are the exact solutions for the corresponding time steps t_k.

Characteristic for the description of cooperative UAV systems as considered in this thesis are logical and mixed-logical conditions, the modeling of which requires binary indicator variables.

Examples of such conditions in the multi-UAV context are:

a) *"A checkpoint is visited by a UAV if the distance between checkpoint and UAV is less than 5 m."*

b) *"If UAV i is at checkpoint A / following target B / ..., then no other UAV is required to do so."*

c) *"UAVs should only minimize distances to unvisited checkpoints / unobserved targets / ..."*

d) *"At least one UAV has to be within communication range to the ground station."*

All the conditions are composed of *propositions* and different *connectives* between them. In Example a), the proposition X_1 'Checkpoint has been visited.' is true if and only if (iff) the proposition X_2 'The UAV-checkpoint distance is less than 5 m.' is true. This would be written as $X_1 \Leftrightarrow X_2$ in Boolean algebra notation. Other possible connectives are $\vee$ (inclusive 'or'), $\wedge$ ('and'), $\sim$ ('not'), $\Rightarrow$ ('if ... then'), and $\oplus$ (exclusive 'or').

A translation of logical statements into linear inequalities involving binary variables is presented in [150]: Let X_i denote the proposition $[\delta_i = 1]$, where δ_i is an *indicator variable* $\in \{0,1\}$. Then it can be seen that

$$X_1 \vee X_2 \quad \text{is equivalent to} \quad \delta_1 + \delta_2 \geq 1, \tag{2.13a}$$
$$X_1 \wedge X_2 \quad \text{is equivalent to} \quad \delta_1 = 1, \delta_2 = 1, \tag{2.13b}$$
$$\sim X_1 \quad \text{is equivalent to} \quad \delta_1 = 0, \tag{2.13c}$$
$$X_1 \Rightarrow X_2 \quad \text{is equivalent to} \quad \delta_1 - \delta_2 \leq 0, \tag{2.13d}$$
$$X_1 \Leftrightarrow X_2 \quad \text{is equivalent to} \quad \delta_1 - \delta_2 = 0, \tag{2.13e}$$
$$X_1 \oplus X_2 \quad \text{is equivalent to} \quad \delta_1 + \delta_2 = 1. \tag{2.13f}$$

However, statements like Example a) combine logical propositions with those concerning a vehicle's motion dynamics, i.e. propositions like $[h(\mathbf{x}) \leq 0]$ that depend on continuous variables $\mathbf{x} \in \mathbb{R}^{n_x}$. The coupling of continuous and logical variables can be represented by mixed-integer linear inequalities as described in the following (based on [150, 15]).

Let $h : \mathbb{R}^{n_x} \to \mathbb{R}$ be linear and $\mathbf{x} \in \mathcal{X}$ with $\mathcal{X} \subset \mathbb{R}^{n_x}$ a given bounded set. Then

$$[h(\mathbf{x}) \leq 0] \vee [\delta = 1] \quad \text{is true iff} \quad h(\mathbf{x}) \leq M\delta, \tag{2.14a}$$
$$[h(\mathbf{x}) \leq 0] \wedge [\delta = 1] \quad \text{is true iff} \quad h(\mathbf{x}) - \delta \leq -1 + m(1-\delta), \tag{2.14b}$$
$$\sim [h(\mathbf{x}) \leq 0] \quad \text{is true iff} \quad h(\mathbf{x}) \geq \varepsilon, \tag{2.14c}$$
$$[h(\mathbf{x}) \leq 0] \Rightarrow [\delta = 1] \quad \text{is true iff} \quad h(\mathbf{x}) \geq \varepsilon + (m-\varepsilon)\delta, \tag{2.14d}$$
$$[h(\mathbf{x}) \leq 0] \Leftrightarrow [\delta = 1] \quad \text{is true iff} \quad \begin{cases} h(\mathbf{x}) & \leq & M(1-\delta), \\ h(\mathbf{x}) & \geq & \varepsilon + (m-\varepsilon)\delta, \end{cases} \tag{2.14e}$$
$$[h(\mathbf{x}) \leq 0] \oplus [\delta_1 = 1] \quad \text{is true iff} \quad \begin{cases} h(\mathbf{x}) & \leq & M(1-\delta_2), \\ h(\mathbf{x}) & \geq & \varepsilon + (m-\varepsilon)\delta_2, \\ \delta_1 + \delta_2 & = & 1. \end{cases} \tag{2.14f}$$

In the above equations, ε represents a small tolerance close to machine precision and

$$M := \max_{\mathbf{x}\in\mathcal{X}} h(\mathbf{x}), \tag{2.15}$$

$$m := \min_{\mathbf{x}\in\mathcal{X}} h(\mathbf{x}). \tag{2.16}$$

The modeling technique is therefore typically referred to as *Big-M* method.

The translation of products into linear mixed-integer constraints requires the introduction of auxiliary variables:

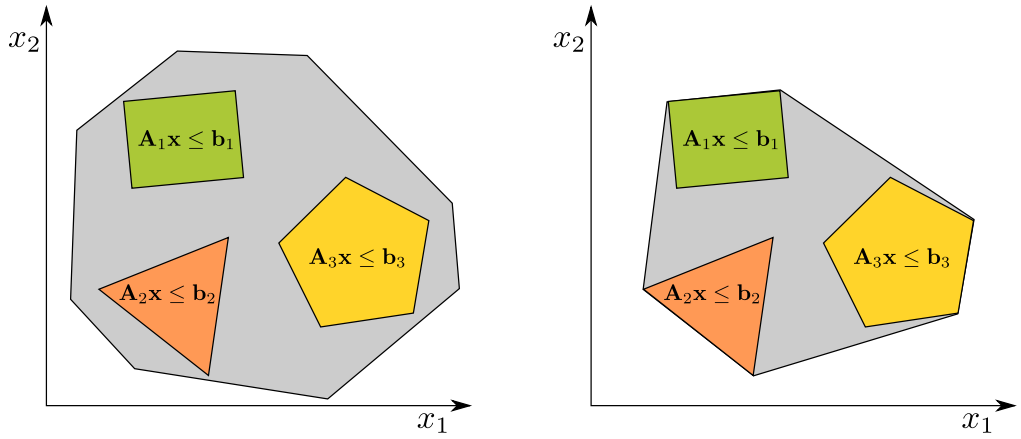
For a product of binary variables $\delta_1\delta_2$, introduce the auxiliary binary variable δ_3, such that

$$\delta_3 = \delta_1\delta_2, \text{ which is equivalent to } \begin{cases} -\delta_1 + \delta_3 \leq 0, \\ -\delta_2 + \delta_3 \leq 0, \\ \delta_1 + \delta_2 - \delta_3 \leq 1. \end{cases} \tag{2.17}$$

For a product of continuous and logical variables $\delta h(\mathbf{x})$, introduce the auxiliary continuous variable z, such that

$$z = \delta h(\mathbf{x}), \text{ which is equivalent to } \begin{cases} z \leq M\delta, \\ z \geq m\delta, \\ z \leq h(\mathbf{x}) - m(1-\delta), \\ z \geq h(\mathbf{x}) - M(1-\delta). \end{cases} \tag{2.18}$$

A considerable alternative to Big-M formulations is the concept of *convex hulls*, originating in the field of Generalized Disjunctive Programming [57]. The main difference between Big-M and convex hull formulations is the tightness of the resulting integer relaxation as illustrated in Figure 2.1.



(a) Big-M relaxation. **(b)** Convex hull relaxation.

Figure 2.1: Example of a disjunctive feasible region with **(a)** Big-M relaxation and **(b)** convex hull relaxation.

The derivation of tight formulations is an important issue in mixed-integer programming since the tighter the relaxation, the faster the solution. Recently, the authors of [143] proposed an

improved Big-M reformulation for increased tightness, but convex hull relaxations are still the tightest one can get. However, in contrast to convex hull formulations, Big-M formulations ensure equivalence with the original expression and require fewer variables and constraints, which needs to be traded off when selecting a suitable modeling technique. In addition, the lack of efficiency can further be reduced by choosing m and M as realistically as possible.

2.3.3 Hybrid Systems Control by Mixed Integer Programming

For optimal control of discrete-time hybrid systems, the MLD model (2.2) is combined with an objective function to form a Constrained Finite Time Optimal Control (CFTOC) Problem:

$$\min_{U_N} \quad \|\mathbf{P}\mathbf{x}^N\|_p + \sum_{k=0}^{N-1} \|\mathbf{Q}_1\mathbf{u}^k\|_p + \|\mathbf{Q}_2\boldsymbol{\delta}^k\|_p + \|\mathbf{Q}_3\mathbf{z}^k\|_p + \|\mathbf{Q}_4\mathbf{x}^k\|_p \tag{2.19a}$$

$$\text{s.t.} \quad \mathbf{x}^{k+1} = \mathbf{A}\mathbf{x}^k + \mathbf{B}_1\mathbf{u}^k + \mathbf{B}_2\boldsymbol{\delta}^k + \mathbf{B}_3\mathbf{z}^k \tag{2.19b}$$

$$\mathbf{y}^k = \mathbf{C}\mathbf{x}^k + \mathbf{D}_1\mathbf{u}^k + \mathbf{D}_2\boldsymbol{\delta}^k + \mathbf{D}_3\mathbf{z}^k \tag{2.19c}$$

$$\mathbf{E}_2\boldsymbol{\delta}^k + \mathbf{E}_3\mathbf{z}^k \leq \mathbf{E}_1\mathbf{u}^k + \mathbf{E}_4\mathbf{x}^k + \mathbf{E}_5 \tag{2.19d}$$

$$\mathbf{x}^N \in \mathcal{X}_f, \tag{2.19e}$$

where $p \in \{1, 2, \infty\}$. The sequence of control inputs $U_N := \{\mathbf{u}^k\}_{k=0}^{N-1}$ is the optimization variable and $\mathcal{X}_f$ a compact terminal target set.

For $p = 2$, problem (2.19) can be reformulated as Mixed Integer Quadratic Program (MIQP) and for $p = 1$ or $p = \infty$ as MILP of the general form

$$\min_{\bar{\mathbf{x}}} \quad \mathbf{r}^T\bar{\mathbf{x}} \tag{2.20a}$$

$$\text{s.t.} \quad \mathbf{H}\bar{\mathbf{x}} \leq \mathbf{w}, \tag{2.20b}$$

where $\bar{\mathbf{x}} := (\bar{\mathbf{x}}_c^T, \bar{\mathbf{x}}_b^T)^T$ combines continuous and binary variables and the vectors $\mathbf{r}$, $\mathbf{w}$, and matrix $\mathbf{H}$ are of suitable size [18].

Mixed Integer Programs (MIPs) are non-convex and $\mathcal{NP}$-complete problems, i.e. in the worst case, the solution time depends exponentially on the number of discrete variables. However, they can always be solved to the global optimum.

By now, highly efficient solvers exist, Gurobi [59] being the fastest, followed by CPLEX [68] (according to recent benchmarks presented in [58]). They typically employ *Branch & Bound* or *Branch & Cut* enumeration techniques to search the whole feasible region for the global optimum. The initialization of these methods relies on the Linear Programming (LP) relaxation of the problem, obtained by constraining δ to be in $[0, 1]$ instead of $\{0, 1\}$. The LP solution provides a lower bound for the MIP solution (*Bounding*). The search space is then iteratively restricted by additional constraints (*Branching*) until the global optimum is found. This explains that the tightness of the relaxation as already mentioned in Section 2.3.2 has a significant influence on the efficiency of the solution process as it determines the size of the search space and therefore the required enumeration effort.

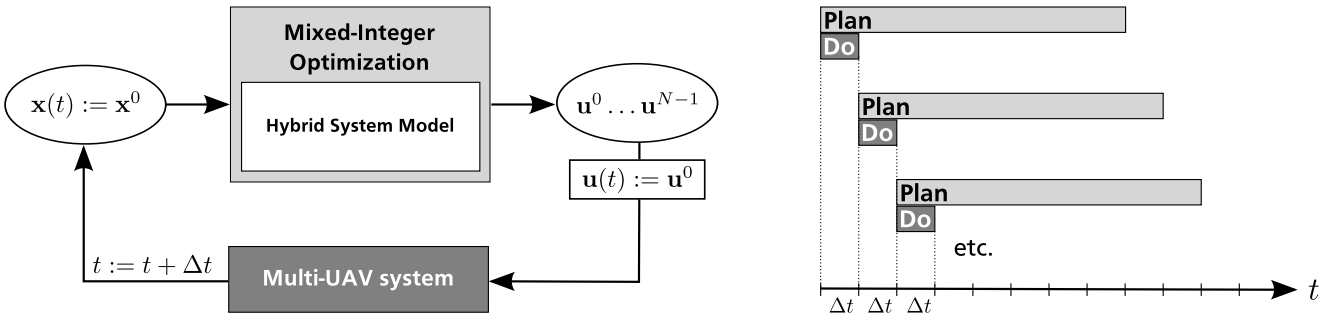
For cooperative mobility problems modeled as CFTOC problem (2.19), its solution U_N represents sequences of control inputs for all involved vehicles that, applied successively, determine vehicle trajectories that are optimal with respect to the modeled common mission objective.

2.3.4 Model-Predictive Control of Linear Hybrid Systems

The problem (2.19) can be solved for finite time horizons N to obtain feedforward vehicle control. By repeatedly solving the problem for smaller N shifted over time, an infinite-time feedback controller is obtained as will be described in the following. Parts of this section were adopted from [78], the definitions and theorems are based on [15].

Model-Predictive Control (MPC), equivalently referred to as Receding Horizon Control (RHC), is an optimal control strategy for (nonlinear) constrained systems. In this thesis, MPC of discrete-time systems will be considered only. The basic idea is to use a model of the system and at each sampling time solve an optimization problem, which predicts the optimal state evolution over a finite time horizon N according to some optimality criterion. N is referred to as *prediction horizon* and is typically chosen significantly smaller than the considered overall number T of time steps of size Δt. The output of the optimization procedure is a sequence of optimal control inputs $u^0, \ldots, u^{N-1}$. The first element of the sequence is then applied to the system and its state at the next sampling time is observed. The procedure is repeated with the new system state, see Figure 2.2. In this manner, the prediction horizon N is shifted or *receded* over time.

Notation: In the following, $\mathbf{x}(t)$ and $\mathbf{u}(t)$, $t \in [t_0, t_0 + T \cdot \Delta t]$, denote the states and control inputs at the global time t, while $\mathbf{x}^0, \ldots, \mathbf{x}^k, \ldots, \mathbf{x}^N$ and $\mathbf{u}^0, \ldots, \mathbf{u}^k, \ldots, \mathbf{u}^{N-1}$ represent the states and control inputs within the scope of the optimization problem for prediction horizon N, i.e. $\mathbf{x}^0 = \mathbf{x}(t)$, $\mathbf{x}^k = \mathbf{x}(t + k \cdot \Delta t)$, and $\mathbf{u}^0 = \mathbf{u}(t)$, $\mathbf{u}^k = \mathbf{u}(t + k \cdot \Delta t)$.



(a) MPC loop. **(b)** RHC policy.

Figure 2.2: Schematic views illustrating the model-predictive feedback control loop **(a)** and the receding horizon control policy **(b)**, respectively.

The most obvious advantage of this strategy is the ability to compensate modeling errors or external disturbances of the system. It has been successfully applied to systems with large sampling times, e.g. chemical processes, where the time-span for solving the optimization problem is not limited [32]. Compared to solving an overall optimization problem for all T time steps, the optimization procedure in receding horizon fashion is clearly less complex. Assuming an exponential dependence on the number of time steps, the solution of one optimization problem over the time horizon T would have complexity 2^T, while the model-predictive approach would have complexity $T2^N$. For short prediction horizons N, this is an important advantage over open-loop optimal control strategies.

Important features of feedback controllers, like stability, robustness, convergence, and feasibility, are in MPC established by recursion [121]. The stabilizing effect of MPC on MLD systems is detailed in the following paragraph.

Stability

Standard stability definitions can be adjusted to fit into the MLD framework (2.2).

Let $\mathbb{X} \times \mathbb{U} \subset \mathbb{R}^{n_x+n_u}$ be a bounded set of feasible states $\mathbf{x}$ and control inputs $\mathbf{u}$ satisfying all physical constraints on the system components, such that

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix} \in \mathbb{X} \times \mathbb{U} := \left\{ \begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix} \in \mathbb{R}^{n_x+n_u} : \mathbf{K}_x \mathbf{x} + \mathbf{K}_u \mathbf{u} \leq \mathbf{L} \right\} . \tag{2.21}$$

Assume that these constraints are included in (2.2c).

Definition 2.3.4. ***Equilibrium pair [15].*** A vector $\mathbf{x}_e \in \mathbb{R}^{n_x^c} \times \{0,1\}^{n_x^b}$ is said to be an equilibrium state for (2.2) and input $\mathbf{u}_e \in \mathbb{R}^{n_u^c} \times \{0,1\}^{n_u^b}$ if $\left(\begin{smallmatrix} \mathbf{x}_e \\ \mathbf{u}_e \end{smallmatrix}\right) \in \mathbb{X} \times \mathbb{U}$ and $\mathbf{x}(k, k_0, \mathbf{x}_e, \mathbf{u}_e) = \mathbf{x}_e$, $\forall k \geq k_0$, $\forall k_0 \in \mathbb{Z}$. The pair $(\mathbf{x}_e, \mathbf{u}_e)$ is said to be an *equilibrium pair*.

$\mathbf{x}(k, k_0, \mathbf{x}_e, \mathbf{u}_e)$ is the system state that results from $(k - k_0)$ evaluations of (2.2) with $\mathbf{u}^k = \mathbf{u}_e$ starting from the initial state $\mathbf{x}^{k_0} = \mathbf{x}_e$.

Definition 2.3.5. ***Stability [15].*** Given an equilibrium pair $(\mathbf{x}_e, \mathbf{u}_e)$, $\mathbf{x}_e \in \mathbb{R}^{n_x^c} \times \{0,1\}^{n_x^b}$ is said to be *stable* if, given $k_0 \in \mathbb{Z}$, $\forall \varepsilon > 0$ $\exists\, \mu(\varepsilon, k_0)$ such that $\|\mathbf{x}^0 - \mathbf{x}_e\| \leq \mu \Rightarrow \|\mathbf{x}(k, k_0, \mathbf{x}^0, \mathbf{u}_e) - \mathbf{x}_e\| \leq \varepsilon$, $\forall k \geq k_0$.

Definition 2.3.6. ***Asymptotic stability [15].*** Given an equilibrium pair $(\mathbf{x}_e, \mathbf{u}_e)$, $\mathbf{x}_e \in \mathbb{R}^{n_x^c} \times \{0,1\}^{n_x^b}$ is said to be *asymptotically stable* if $\mathbf{x}_e$ is stable and $\exists r > 0$ such that $\forall \mathbf{x}^0 \in \mathcal{B}(\mathbf{x}_e, r)$ and $\forall \varepsilon > 0$ $\exists K(\varepsilon, k_0)$ such that $\|\mathbf{x}(k, k_0, \mathbf{x}^0, \mathbf{u}_e) - \mathbf{x}_e\| \leq \varepsilon$, $\forall k \geq K$.

For the binary component $\mathbf{x}_b$ of the state vector, Definition 2.3.6 means that there exists a finite time $\bar{k}$ such that $\mathbf{x}_b^k \equiv \mathbf{x}_{be}$, $\forall k \geq \bar{k}$. That permits to consider *local stability* only to depend on the continuous part $\mathbf{x}_c$ of the state vector. In particular, there exists a neighborhood of $\mathbf{x}_{ce}$ in which $\mathbf{x}_c^k$ can be perturbed without violating $\mathbf{x}_b^k = \mathbf{x}_{be}$.

Given an equilibrium pair $(\mathbf{x}_e, \mathbf{u}_e)$, the corresponding values of well-posed components of the auxiliary vectors $\mathbf{z}$ and $\boldsymbol{\delta}$ can be determined via the functions $\mathcal{Z}_i$ and $\mathcal{D}_i$ introduced in Definition 2.3.2. In order to allow for indefinite components as well, the following definition is useful.

Definition 2.3.7. ***Definite admissibility [15].*** Let $(\mathbf{x}_e, \mathbf{u}_e)$ be an equilibrium pair for a time-invariant MLD system, and let the system be well-posed by Definition 2.3.2. For $i \in \mathcal{I}$, $j \in \mathcal{J}$, let $\delta_{e,i}$, $z_{e,j}$ be the corresponding equilibrium auxiliary variables. An auxiliary vector $\boldsymbol{\delta}$ (or $\mathbf{z}$) is said to be *definitely admissible* if $\delta_i = \delta_{e,i}$, $\forall i \in \mathcal{I}$, $(z_j = z_{e,j}, \forall j \in \mathcal{J})$, and

$$\mathbf{E}_2 \boldsymbol{\delta} + \mathbf{E}_3 \mathbf{z} \leq \mathbf{E}_1 \mathbf{u}_e + \mathbf{E}_4 \mathbf{x}_e + \mathbf{E}_5 . \tag{2.22}$$

MPC stabilizes MLD and PWA systems, respectively, or drives them to desired reference trajectories. Under certain assumptions, stability can even be guaranteed, as stated in the following theorem.

Theorem 2.3.1. [18] Let $(\mathbf{x}_e, \mathbf{u}_e) = (\mathbf{0}, \mathbf{0})$ be an equilibrium pair and let $(\boldsymbol{\delta}_e, \mathbf{z}_e) = (\mathbf{0}, \mathbf{0})$ be definitely admissible. Assume that the initial state $\mathbf{x}(0)$ is such that a feasible solution of problem (2.19) with $\mathcal{X}_f = \mathbf{0}$ and $\mathbf{x}^0 = \mathbf{x}(t)$ exists at time $t = 0$. Then for all $\mathbf{Q}_1 = \mathbf{Q}_1^T \succ 0$, $\mathbf{Q}_2 = \mathbf{Q}_2^T \succeq 0$, $\mathbf{Q}_3 = \mathbf{Q}_3^T \succeq 0$, and $\mathbf{Q}_4 = \mathbf{Q}_4^T \succ 0$ if $p = 2$ (for all $\mathbf{Q}_1$ and $\mathbf{Q}_4$ of full column rank if $p = 1, \infty$), the mixed-integer predictive control law obtained by repeatedly solving (2.19) with $\mathbf{x}^0 = \mathbf{x}(t)$ and applying $\mathbf{u}(t) = \mathbf{u}^0$ at each time t according to the RHC policy stabilizes the system in that it converges to the equilibrium, i.e.

$$\begin{aligned}
&\lim_{t\to\infty} \mathbf{x}(t) = 0 \,, \\
&\lim_{t\to\infty} \mathbf{u}(t) = 0 \,, \\
&\lim_{t\to\infty} \|\mathbf{Q}_2 \boldsymbol{\delta}(t)\|_p = 0 \,, \\
&\lim_{t\to\infty} \|\mathbf{Q}_3 \mathbf{z}(t)\|_p = 0 \,.
\end{aligned}$$

If $\mathbf{Q}_2 \succ 0$ and $\mathbf{Q}_3 \succ 0$ for $p = 2$ or if $\mathbf{Q}_2$ and $\mathbf{Q}_3$ are full column rank matrices for $p = 1, \infty$, then also $\boldsymbol{\delta}(t)$ and $\mathbf{z}(t)$ converge.

Prove of stability traded off against controller efficiency

It has to be noted, that the MILPs within the mixed-integer predictive control law do not have to be solved to their global optimum in order for the MPC to be stabilizing. It is, therefore, possible to introduce a time limit for the MILP solution and work with suboptimal solutions only. That way, certain efficiency bounds can be guaranteed.

In most applications, not all matrices $\mathbf{Q}_i$ satisfy the requirements of Theorem 2.3.1 or shall explicitly be left as free tuning parameters. In that case, for non-singular matrices $\mathbf{Q}_i$, stability can be guaranteed by introducing certain constraints on the terminal state $\mathbf{x}^N$ or by appropriately choosing the terminal weight $\mathbf{P}$ (cf. [12, 13, 18, 80]). In the vehicle control context, constraints on the terminal state ensure that feasible vehicle motion even beyond the prediction horizon is possible. However, such a terminal state constraint might restrict the controller [121] or modeling the terminal set $\mathcal{X}_f$ might require many inequalities that increase the problem complexity and prohibit time-critical MPC applications [134]. Therefore, the authors of [134] investigated the practical necessity of a terminal state constraint for stability and observed that MPC can be stabilizing even without terminal constraints. In order to still have an indication of the stability of a model-predictive controller, they proposed an a-posteriori stability analysis tool.

Well-posedness of the MLD system is another condition for Theorem 2.3.1 to hold. Systems that are not well-posed in the first place can be made well-posed by adding additional constraints to the model. This might also increase the MILP solution time and has to be traded off against the desired efficiency.

2.4 Proposed Mixed-Integer MPC Scheme for Decentralized Cooperative UAV Control

The concept of mixed-integer MPC can be adopted for the control of multiple vehicles that are to cooperatively perform a mobility task. The logical part of the mixed-logical model then

typically represents the allocation of target locations in the workspace to the different members of the vehicle team. Mixed-logical constraints are also necessary for modeling distance-based relations, e.g. collision or obstacle avoidance. They might as well result from the approximation of nonlinear model parts, i.e. from hybridization as introduced in Section 2.3.2. The continuous model part essentially represents the motion dynamics of the vehicles, but can also describe other continuous processes like time-dependent penalty increase.

This section presents the vehicle dynamics models adopted to represent quadrotor and fixed-wing UAV motion in the application scenarios in Chapter 4 as well as the employed approach for linearizing Euclidean Distances. These are basic elements recurring in every MPC application considered in this thesis. Moreover, details on the proposed decentralization of the mixed-integer MPC scheme will be given.

Selected application-specific modeling details along with the description of the respective application scenario can be found in Sections 2.5, 4.5, and 4.6.

2.4.1 Linear Approximation of Vehicle Dynamics Models

Basic Quadrotor Model

Quadrotor UAVs are modeled as point mass with double integrator dynamics $\ddot{\mathbf{x}} = \mathbf{u}$, i.e. the first order Ordinary Differential Equation (ODE) describing the motion dynamics is

$$\dot{\mathbf{x}}_{qr}(t) = \begin{pmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \\ u_x(t) \\ u_y(t) \\ u_z(t) \end{pmatrix} = \mathbf{f}_{qr}(\mathbf{x}_{qr}(t), \mathbf{u}_{qr}(t)) \,, \tag{2.23}$$

where the state vector $\mathbf{x}_{qr}(t) = (x(t), y(t), z(t), v_x(t), v_y(t), v_z(t))^T$ contains the quadrotor's x/y/z positions and velocities, and the vector of control inputs $\mathbf{u}_{qr}(t) = (u_x(t), u_y(t), u_z(t))^T$) comprises its x/y/z accelerations. The x/y/z coordinates correspond to the world reference frame as illustrated in Figure 2.3. It is also possible to extend $\mathbf{f}_{qr}$ to account for disturbances, e.g. induced by wind.

Euler discretization (as per (2.10)) of (2.23) yields

$$\mathbf{x}_{qr}^{k+1} = \mathbf{x}_{qr}^{k} + \Delta t \cdot \mathbf{f}_{qr}(\mathbf{x}_{qr}^{k}, \mathbf{u}_{qr}^{k}) \,, \tag{2.24}$$

where Δt is the step size and the superscript k relates to the time step $t_k = k \cdot \Delta t$.

In order to derive the exact discretization (as per (2.12)) of the quadrotor dynamics, (2.23) is rewritten as continuous-time affine state space model

$$\dot{\mathbf{x}}_{qr}(t) = \mathbf{A}^C \mathbf{x}_{qr}(t) + \mathbf{B}^C \mathbf{u}_{qr}(t) + \mathbf{a}^C \tag{2.25}$$

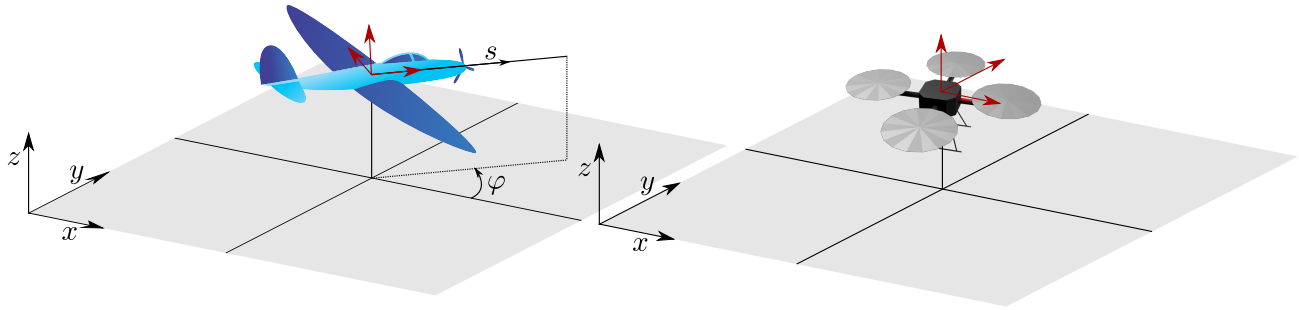


Figure 2.3: Simple airplane and quadrotor model. In contrast to the omnidirectional point mass model of the quadrotor, the airplane model does reflect the orientation of the UAV.

with

$$\mathbf{A}^C = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad \mathbf{B}^C = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{and} \quad \mathbf{a}^C = \mathbf{0}\,.$$

From that results the exact discretization

$$\mathbf{x}_{qr}^{k+1} = \mathbf{A}^D \mathbf{x}_{qr}^k + \mathbf{B}^D \mathbf{u}_{qr}^k + \mathbf{a}^D\,, \tag{2.26}$$

where

$$\mathbf{A}^D = \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{B}^D = \begin{pmatrix} \frac{\Delta t^2}{2} & 0 & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{pmatrix}, \quad \text{and} \quad \mathbf{a}^D = \mathbf{0}\,.$$

Basic Airplane Model

A basic second-order airplane model [82] is used to describe the motion of fixed-wing UAVs:

$$\dot{\mathbf{x}}_{fw}(t) = \begin{pmatrix} s \cdot \cos\varphi(t) \\ s \cdot \sin\varphi(t) \\ v_z(t) \\ \omega_\varphi(t) \\ u_z(t) \\ u_\omega(t) \end{pmatrix} = \mathbf{f}_{fw}(\mathbf{x}_{fw}(t), \mathbf{u}_{fw}(t))\,. \tag{2.27}$$

Here, the state vector $\mathbf{x}_{fw}(t) = (x(t), y(t), z(t), \varphi(t), v_z(t), \omega_\varphi(t))^T$ contains the UAV's x/y/z position, orientation φ, climb/descent rate v_z, and angular velocity ω_φ. Control inputs

$\mathbf{u}_{fw}(t) = (u_z(t), u_\omega(t))^T$ are the climb/descent acceleration u_z and the angular acceleration u_ω. A constant forward speed $s = const$ is assumed. Figure 2.3 illustrates the state description. Takeoff and landing of the aircraft are not modeled.

Euler discretization of (2.27) yields

$$\mathbf{x}_{fw}^{k+1} = \mathbf{x}_{fw}^{k} + \Delta t \cdot \mathbf{f}_{fw}(\mathbf{x}_{fw}^{k}, \mathbf{u}_{fw}^{k}) \,. \tag{2.28}$$

Equation (2.28) still contains the nonlinearities $\sin\varphi$ and $\cos\varphi$. For incorporating the fixed-wing UAV dynamics into problem (2.19) or (2.20), respectively, a piecewise affine state-space representation is required. It can be derived by solving the Nonlinear Program (NLP) (2.8) for optimal hybridization of the nonlinear model part

$$\hat{\mathbf{f}} : [-\pi, \pi] \to [-1, 1]^2 \quad \text{with} \quad \hat{\mathbf{f}}(\varphi) = \begin{pmatrix} \sin\varphi \\ \cos\varphi \end{pmatrix} .$$

For this purpose, the interval $[-\pi, \pi]$ was divided into $n = 8$ partitions $\mathcal{S}_i = [\varphi_l^i, \varphi_u^i)$, $i = 1, 2, \ldots, n$. On each of the partitions, $\hat{\mathbf{f}}$ is approximated by the linear expression

$$\hat{\mathbf{f}}(\varphi) \approx \hat{\mathbf{f}}_{\text{lin}}^i(\varphi) = \begin{pmatrix} m_{\sin}^i \\ m_{\cos}^i \end{pmatrix} \cdot \varphi + \begin{pmatrix} c_{\sin}^i \\ c_{\cos}^i \end{pmatrix} \quad \text{if} \quad \varphi \in \mathcal{S}_i \,, \tag{2.29}$$

where

$$m_{\sin}^i = \frac{z_{\sin}^{i,u} - z_{\sin}^{i,l}}{\varphi_u^i - \varphi_l^i} \,, \qquad c_{\sin}^i = z_{\sin}^{i,l} - m_{\sin}^i \cdot \varphi_l^i \,, \qquad \begin{pmatrix} z_{\sin}^{i,l} \\ z_{\cos}^{i,l} \end{pmatrix} = \hat{\mathbf{f}}_{\text{lin}}^i(\varphi_l^i) \,,$$

$$m_{\cos}^i = \frac{z_{\cos}^{i,u} - z_{\cos}^{i,l}}{\varphi_u^i - \varphi_l^i} \,, \qquad c_{\cos}^i = z_{\cos}^{i,l} - m_{\cos}^i \cdot \varphi_l^i \,, \qquad \begin{pmatrix} z_{\sin}^{i,u} \\ z_{\cos}^{i,u} \end{pmatrix} = \hat{\mathbf{f}}_{\text{lin}}^i(\varphi_u^i) \,.$$

Due to the periodicity of sin and cos, the constraints $z_{\sin}^{1,l} = z_{\sin}^{8,u}$ and $z_{\cos}^{1,l} = z_{\cos}^{8,u}$ were added to the optimization problem (2.8). The solution values for φ_l^i, φ_u^i and $\hat{\mathbf{f}}_{\text{lin}}^i(\varphi_l^i), \hat{\mathbf{f}}_{\text{lin}}^i(\varphi_u^i)$ are summarized in Table 2.1. The resulting piecewise affine approximation is shown in Figure 2.4.

After hybridization, the fixed-wing UAV dynamics can be written as continuous-time PWA system

$$\dot{\mathbf{x}}_{fw}(t) = \mathbf{A}_i^C \mathbf{x}_{fw}(t) + \mathbf{B}_i^C \mathbf{u}_{fw}(t) + \mathbf{a}_i^C \quad \text{if} \quad \varphi(t) \in \mathcal{S}_i \,, i = 1, 2, \ldots, n \tag{2.30}$$

with

$$\mathbf{A}_i^C = \begin{pmatrix} 0 & 0 & 0 & s \cdot m_{\cos}^i & 0 & 0 \\ 0 & 0 & 0 & s \cdot m_{\sin}^i & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{B}_i^C = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \text{and} \quad \mathbf{a}_i^C = \begin{pmatrix} s \cdot c_{\cos}^i \\ s \cdot c_{\sin}^i \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .$$

	φ_l^1	$\varphi_u^1 = \varphi_l^2$	$\varphi_u^2 = \varphi_l^3$	$\varphi_u^3 = \varphi_l^4$	$\varphi_u^4 = \varphi_l^5$
φ	-3.1416	-2.5294	-1.8890	-1.1237	-0.3644
$\hat{\mathbf{f}}_{\text{lin}}^i(\varphi) = \hat{\mathbf{f}}_{\text{lin}}^{i+1}(\varphi)$	$\begin{pmatrix} -0.9942 \\ 0.0012 \end{pmatrix}$	$\begin{pmatrix} -0.8520 \\ -0.5945 \end{pmatrix}$	$\begin{pmatrix} -0.3232 \\ -0.9824 \end{pmatrix}$	$\begin{pmatrix} 0.4517 \\ -0.9477 \end{pmatrix}$	$\begin{pmatrix} 0.9798 \\ -0.3733 \end{pmatrix}$

	$\varphi_u^5 = \varphi_l^6$	$\varphi_u^6 = \varphi_l^7$	$\varphi_u^7 = \varphi_l^8$	φ_u^8
φ	0.4518	1.2651	2.2449	3.1416
$\hat{\mathbf{f}}_{\text{lin}}^i(\varphi) = \hat{\mathbf{f}}_{\text{lin}}^{i+1}(\varphi)$	$\begin{pmatrix} 0.9497 \\ 0.4635 \end{pmatrix}$	$\begin{pmatrix} 0.3198 \\ 0.9946 \end{pmatrix}$	$\begin{pmatrix} -0.6850 \\ 0.8505 \end{pmatrix}$	$\begin{pmatrix} -0.9942 \\ 0.0012 \end{pmatrix}$

Table 2.1: Solution values corresponding to the optimal hybridization of sin and cos depicted in Figure 2.4.

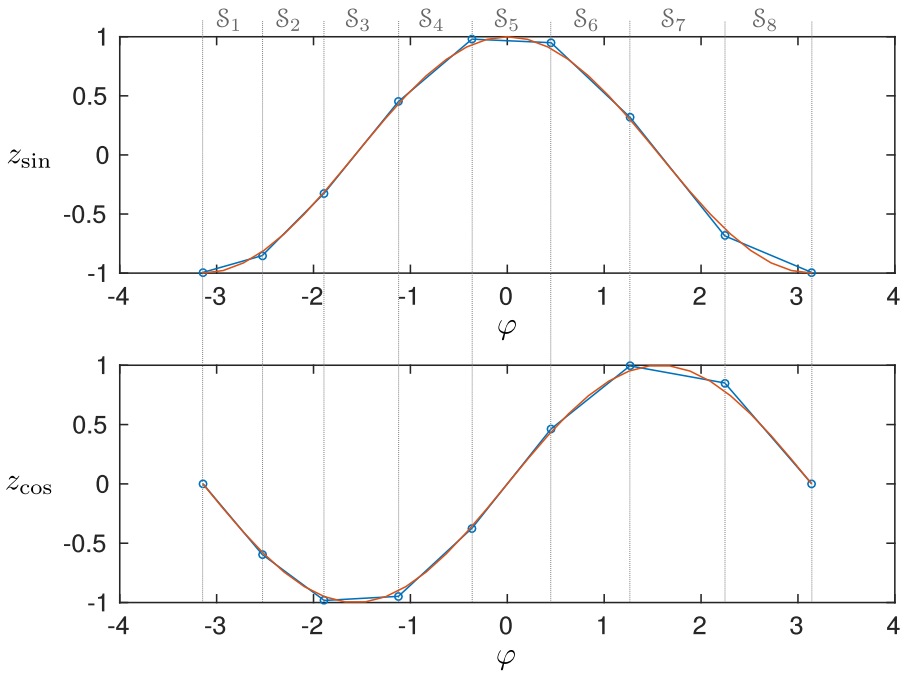


Figure 2.4: Optimal hybridization of sin and cos with $n = 8$ partitions.

Euler discretization of (2.30) yields the discrete-time PWA system

$$\mathbf{x}_{fw}^{k+1} = \mathbf{A}_i^D \mathbf{x}_{fw}^k + \mathbf{B}_i^D \mathbf{u}_{fw}^k + \mathbf{a}_i^D \quad \text{if} \quad \varphi^k \in \mathcal{S}_i \,, i = 1, 2, \ldots, n \,, \tag{2.31}$$

where

$$\mathbf{A}_i^D = \begin{pmatrix} 1 & 0 & 0 & \Delta t \cdot s \cdot m_{\cos}^i & 0 & 0 \\ 0 & 1 & 0 & \Delta t \cdot s \cdot m_{\sin}^i & 0 & 0 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \; \mathbf{B}_i^D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \Delta t & 0 \\ 0 & \Delta t \end{pmatrix}, \; \mathbf{a}_i^D = \begin{pmatrix} \Delta t \cdot s \cdot c_{\cos}^i \\ \Delta t \cdot s \cdot c_{\sin}^i \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

For the exact discretization of (2.30), one obtains the representation (2.31) with matrices

$$\mathbf{A}_i^D = \begin{pmatrix} 1 & 0 & 0 & \Delta t \cdot s \cdot m_{\cos}^i & 0 & \frac{\Delta t^2}{2} \cdot s \cdot m_{\cos}^i \\ 0 & 1 & 0 & \Delta t \cdot s \cdot m_{\sin}^i & 0 & \frac{\Delta t^2}{2} \cdot s \cdot m_{\sin}^i \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \; \mathbf{B}_i^D = \begin{pmatrix} 0 & \frac{\Delta t^3}{6} \cdot s \cdot m_{\cos}^i \\ 0 & \frac{\Delta t^3}{6} \cdot s \cdot m_{\sin}^i \\ \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{pmatrix}, \; \mathbf{a}_i^D = \begin{pmatrix} \Delta t \cdot s \cdot c_{\cos}^i \\ \Delta t \cdot s \cdot c_{\sin}^i \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

2.4.2 Linear Approximation of Euclidean Distances

Since all of the controller applications considered in this thesis involve decision logic that is based on distances between vehicles and target points or vehicles among each other, a linear representation of the Euclidean distance is an essential part of each mixed-integer model.

For illustrating the employed approximation, consider the 2D distance between a vehicle v with position (x_v, y_v) and a target point t located at (x_t, y_t). The exact Euclidean distance between v and t is given by $\tilde{d}_{vt} = \sqrt{(x_v - x_t)^2 + (y_v - y_t)^2}$. This expression can be linearized using the following approximation:

$$\tilde{d}_{vt} \approx \min \left\{ d_{vt} \;\middle|\; (x_v - x_t) \sin \frac{2\pi i}{n_d} + (y_v - y_t) \cos \frac{2\pi i}{n_d} \leq d_{vt}, \; i = 1, \ldots, n_d, \; n_d \in \mathbb{N} \right\}. \tag{2.32}$$

Each of the n_d inequalities describes a hyperplane in $\mathbb{R}^2$. If d_{vt} takes the lowest value, such that all inequalities hold, the resulting approximation can best be illustrated by means of the circle with radius $\tilde{d}_{vt}$ defined by

$$(x - x_t)^2 + (y - y_t)^2 = \tilde{d}_{vt}^2 \,.$$

The set of inequalities (2.32) approximates the circle as an intersection of n_d hyperplanes as shown in Figure 2.5. For the depicted example of the distance between $(x_v, y_v) = (6, 7)$ and

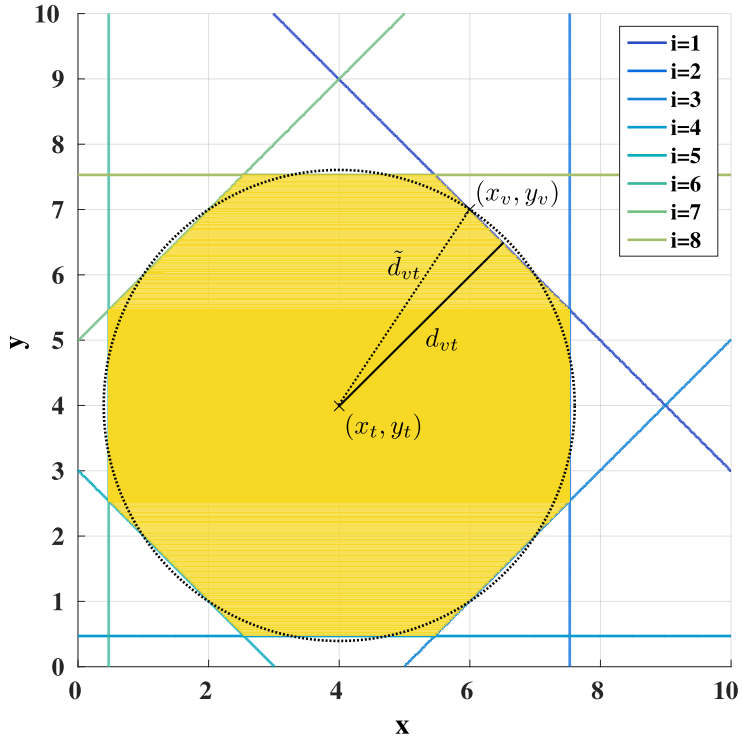


Figure 2.5: Illustration of the linear approximation of the Euclidean distance between $(x_v, y_v) = (6, 7)$ and $(x_t, y_t) = (4, 4)$ using $n_d = 8$ hyperplanes as defined in (2.32).

$(x_t, y_t) = (4, 4)$ using $n_d = 8$ hyperplanes, the exact Euclidean distance is $\tilde{d}_{vt} = 3.6056$ and the approximation yields $d_{vt} = 3.5355$. The accuracy of the approximation can be scaled by the constant parameter $n_d \in \mathbb{N}$.

For describing distances in 3D, additional inequalities have to be introduced, such that d_{vt} has to satisfy

$$(x_v - x_t)\sin\frac{2\pi i}{n_d} + (y_v - y_t)\cos\frac{2\pi i}{n_d} \leq d_{vt}, \tag{2.33a}$$

$$(x_v - x_t)\sin\frac{2\pi i}{n_d} + (z_v - z_t)\cos\frac{2\pi i}{n_d} \leq d_{vt}\,, \quad \text{and} \tag{2.33b}$$

$$(y_v - y_t)\sin\frac{2\pi i}{n_d} + (z_v - z_t)\cos\frac{2\pi i}{n_d} \leq d_{vt}\,, \tag{2.33c}$$

where (x_v, y_v, z_v) denotes the vehicle's position and (x_t, y_t, z_t) the position of some target point. In order to obtain the most accurate values d_{vt} when using the distance approximation in a problem description (2.19) for mixed-integer MPC, not only the constraints (2.33) have to be included in (2.19d), but also the objective function (2.19a) has to comprise the minimization of all d_{vt}.

2.4.3 Decentralization

As already pointed out in Section 2.1, decentrality is a key feature for the control of multi-vehicle systems in settings where communication is restricted or constrained, the team com-

position might change due to vehicle failure or addition, or the computational capacities are limited. Only a decentralized controller is scalable to an arbitrarily large number of vehicles in the system without losing real-time efficiency. This is why in addition to the time decomposition of the overall cooperative control problem (as formulated in (2.19) or (2.20)) by RHC, also a decomposition into local subproblems is proposed here to obtain an even more efficient decentralized cooperative control approach.

The basic idea is to run the model-predictive controller as introduced in Section 2.3.4 on every vehicle individually to provide its optimal next move based only on locally available information on other vehicles. Typically, that means those teammates that are within communication range of the controlled vehicle. An example configuration is shown in Figure 2.6. Here, the controller of UAV s1 would also account for s2 and s3 as they are within communication range indicated by the orange circle. Mathematically speaking, problem (2.19) is set up for the system consisting of UAVs s1, s2, and s3 only, i.e. $\mathbf{x}$ only comprises the state variables and $\mathbf{u}$ the controls of these three vehicles and all other vectors and matrices are adapted to represent the motion dynamics and cooperation logic within the so defined local subsystem. In the same manner, s2's controller would include s1 and s3, s3's controller s1, s2, and s4, and so on.

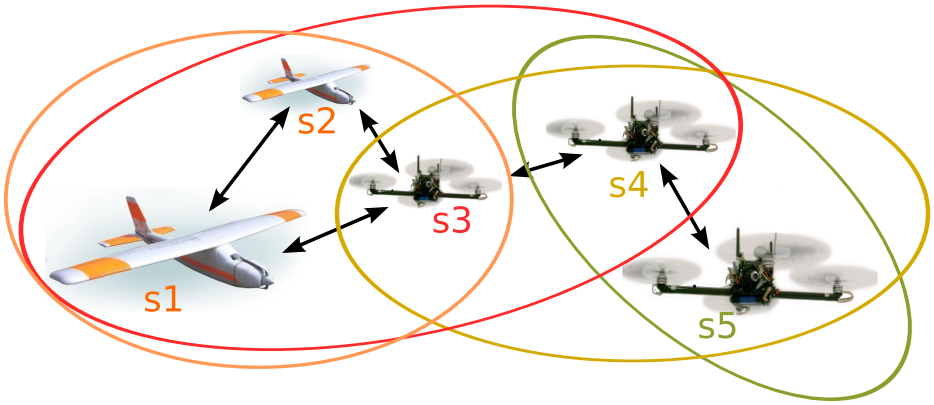


Figure 2.6: Example configuration of a multi-UAV system. The colored circles correspond to the color of the vehicle index and indicate, which teammates are within communication range and would therefore be comprised in the vehicle's local subsystem model used for decentralized control.

This local subsystem model serves as a basis for the MPC predicting the evolution of the system state $\mathbf{x}$ and computing control inputs $\mathbf{u}$ that optimize the objective (2.19a). This means, optimal behavior for all involved vehicles is computed, but only the inputs for the vehicle executing the MPC is actually applied for the next step. Determination of a single vehicle's control is therefore based on the assumption that the movement of all other teammates is optimal within the regarded subsystem. Deviations resulting from the fact that the actual system state differs from its locally predicted evolution are compensated by the MPC strategy as it is rerun every Δt s with the new system state and therefore permanently adapts to the actual system behavior.

Obviously, the general challenge in decentralized cooperative control is that each team member has local information only for deciding on its individual action, which has global impact. Hence, the overall team performance with respect to the global objective employing the decentralized MPC approach can be expected to be inferior to that obtained from a centralized MPC accounting for all vehicles at the same time. The loss in performance depends on the number of vehicles

in each subsystem and if or how much the subsystems overlap. However, as long as an overlap of subsystems similar to that shown in Figure 2.6 can be maintained throughout the mission, cooperation even between vehicles that cannot directly communicate, e.g. s1 and s5, is achieved also by the decentralized MPC. Moreover, it is significantly more efficient than its centralized counterpart since the complexity of problem (2.19) increases exponentially with the number of involved vehicles. By defining an upper bound for the number of vehicles in a subsystem, efficiency of the decentralized MPC can be traded off against cooperative performance.

2.5 Simulation-Based Validation in Different Application Scenarios

The proposed decentralized mixed-integer MPC has been successfully applied and validated in different simulated application scenarios, three examples of which are presented in the following along with the obtained results. Only selected parts of each underlying mixed-integer model that are characteristic for the respective problem representation will be given. A complete model description can be found in the corresponding publications mentioned at the beginning of each subsection.

2.5.1 Cooperative Target Observation

This section summarizes own work published in [79].

Cooperative Multi-Robot Observation of Multiple Moving Targets (CMOMMT) [106] is a key problem in many security, surveillance, and service applications. As it is inherently cooperative and scalable, it still is a well-accepted, $\mathcal{NP}$-hard benchmark problem for investigating situation-based allocation of roles and subtasks as well as the determination of vehicle-specific trajectories [9].

Problem Statement

A fixed number of n_R robots and n_T targets is considered as vehicles in a bounded circular work area $\Omega \subset \mathbb{R}^2$ with no obstacles. A robot is said to *observe* a target if the target is located within the robot's *observation range* with radius $R_1 = 2600$ (cf. Figure 2.7). In addition, a robot is assumed to know about targets within its 360° *sensing range* of radius $R_2 = 3000$, but only if they are observed by some other robot. The overall region covered by the robots' observation sensors is significantly smaller than the considered work area, forcing them to dynamically adjust their movement to nearby targets. It is assumed that the robots' maximum velocity is greater than those of the targets.

The robots are able to communicate information on targets within their sensing range as well as their own position to other teammates. The *range of communication* $R_3 = 5000$ is significantly larger than the observation range but too small to cover the whole work area.

In terms of the decentralized MPC approach as outlined in Section 2.4.3, each robot's individual controller computes its optimized control inputs based on locally available information only, which in the target observation problem comprises all teammates within the robot's communica-

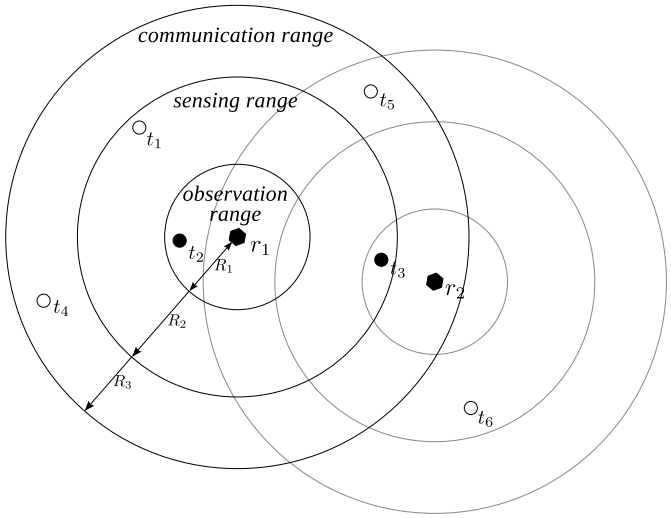


Figure 2.7: Robot r_1 with its observation, sensing, and communication range. r_1 observes target t_2 and knows about t_3. The targets t_1, t_4, t_5, and t_6 are not visible for r_1.

tion range R_3, all targets within its sensing range R_2, and those targets sensed by the teammates it is able to communicate with. The total number of robots and targets in a local subsystem is $\widetilde{n}_R$ and $\widetilde{n}_T$, respectively.

Objective

The common objective is to minimize the total time in which targets escape observation. In order to formalize this goal, the *A-metric* [105] is introduced:

$$A = \frac{1}{n_T} \sum_{k=0}^{T} \sum_{t=1}^{n_T} \frac{a_t(k \cdot \Delta t)}{T} , \qquad \text{(A-metric)}$$

where $a_t(k \cdot \Delta t)$ are binary variables that equal 1 if target t is observed at time $k \cdot \Delta t$ and 0 otherwise. The *A-metric* represents the average percentage of targets being observed by at least one robot at some instant in time throughout a period of T time steps of size Δt. Thus, the mission objective can be restated as the maximization of A, where $A = 1$ represents the maximum value.

Modeling

Robots are modeled as point masses with double integrator dynamics employing exact time discretization, i.e. their motion dynamics model corresponds to a 2D version of (2.26).

Since the work area Ω is considered a circle with radius R_{work}, the model contains the following constraints on the robots' positions, which linearly approximate the circular work area by a polygon with n_d edges:

$$x_r^k \sin \frac{2\pi i}{n_d} + y_r^k \cos \frac{2\pi i}{n_d} \leq R_{work} , \qquad (2.34)$$

where $i = 1, \ldots, n_d$, $k = 0, \ldots, N$, $r = 1, \ldots, \widetilde{n}_R$.

The distances d_{rt} between a robot r and a target t are approximated as described in Section 2.4.2 and used to model the observation condition

$$b_{rt}^k = 1 \quad \Rightarrow \quad d_{rt}^k \leq R_1 \tag{2.35}$$

for $k = 0,\dots,N$, $r = 1,\dots\widetilde{n}_R$, $t = 1,\dots,\widetilde{n}_T$, with $b_{rt}^k \in \{0,1\}$ and $b_{rt}^k = 1$ indicating that robot r observes target t at time step k of the MPC prediction horizon N. Since it is not of interest which robot observes target t, but that it is observed by any team member, another binary variable $s_t^k \in \{0,1\}$ is introduced and represents the general observation status of target t:

$$s_t^k = 0 \quad \Leftrightarrow \quad \sum_{r=1}^{\widetilde{n}_R} b_{rt}^k \geq 1 \tag{2.36}$$

for $k = 0,\dots,N$, $t = 1,\dots,\widetilde{n}_T$.

Cooperation in observing the targets can now be realized by minimizing the number of unobserved targets, i.e. $\sum_{t=1}^{\widetilde{n}_T} s_t^k$, and by minimizing each robot's distance to those targets not yet observed by any other robot, which is expressed using a set of auxiliary variables $h_{rt} \in \mathbb{R}$, such that

$$h_{rt}^k = s_t^k \cdot d_{rt}^k \tag{2.37}$$

with $k = 0,\dots,N$, $r = 1,\dots,\widetilde{n}_R$, $t = 1,\dots,\widetilde{n}_T$. That way, $h_{rt} = 0$ if target t is already observed and $h_{rt} = d_{rt}$ otherwise.

Equations (2.35), (2.36), and (2.37) are transformed into mixed-integer linear expressions employing the Big-M method introduced in Section 2.3.2.

An objective function for the CFTOC problem (2.19) is constructed that minimizes the number of unobserved targets, the distances to unobserved targets, and the required control effort:

$$\min_{U_N} \sum_{k=0}^{N} \Big(q_\delta \sum_{t=1}^{\widetilde{n}_T} s_t^k + q_z \sum_{t=1}^{\widetilde{n}_T} \sum_{r=1}^{\widetilde{n}_R} h_{rt}^k \Big) + q_u \sum_{k=0}^{N-1} \sum_{r=1}^{\widetilde{n}_R} |u_{r,x}^k| + |u_{r,y}^k| \,, \tag{2.38}$$

where the different aspect are weighted by q_δ, q_z, and q_u according to the different objective priorities and the best-expected task performance.

Results

As already mentioned in Section 2.4.3, a maximum number of robots $\widetilde{n}_{Rmax}$ and targets $\widetilde{n}_{Tmax}$ in a model may not be exceeded in order to obtain an efficient decentralized online control strategy. Complexity and calculation time grows exponentially with the model size. Hence, the calculation times for different model sizes were compared in order to find reasonable values for $\widetilde{n}_{Rmax}$ and $\widetilde{n}_{Tmax}$. For this purpose, the MILP representing the cooperative observation mission was set up for different numbers of robots $\widetilde{n}_R$ and targets $\widetilde{n}_T$ as well as different lengths of prediction horizons N. Figure 2.8 gives an impression of the computing times needed to solve a single MILP describing a system of $\widetilde{n}_R = 1,\dots,5$ robots and $2 \cdot \widetilde{n}_R$ targets over a prediction horizon $N = 3$ and $N = 5$, respectively. The boxplots were obtained from 200 solver calls for randomly generated system states for each instantiation of $\widetilde{n}_R$, $\widetilde{n}_T$, and N, performed on a Dual

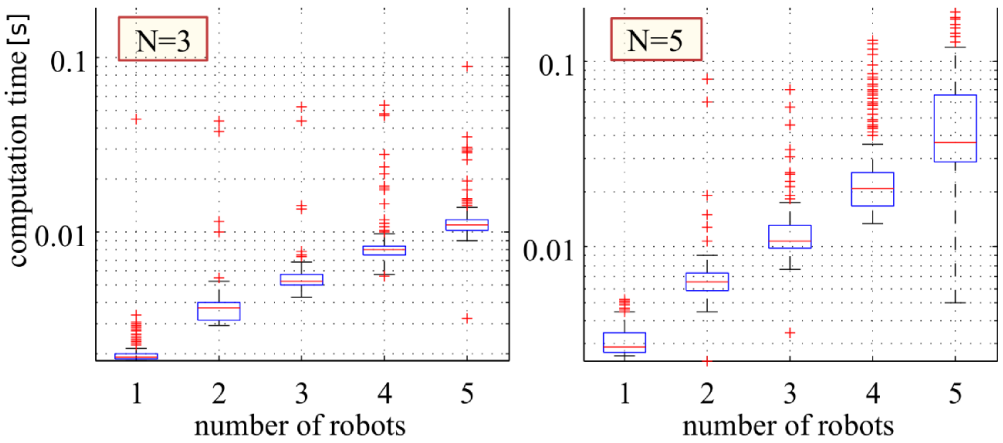


Figure 2.8: Boxplots of the computing times needed to solve a single mixed-integer linear program over prediction horizon $N = 3$ and $N = 5$, respectively, for systems of $\widetilde{n}_R$ robots and $2\,\widetilde{n}_R$ targets.

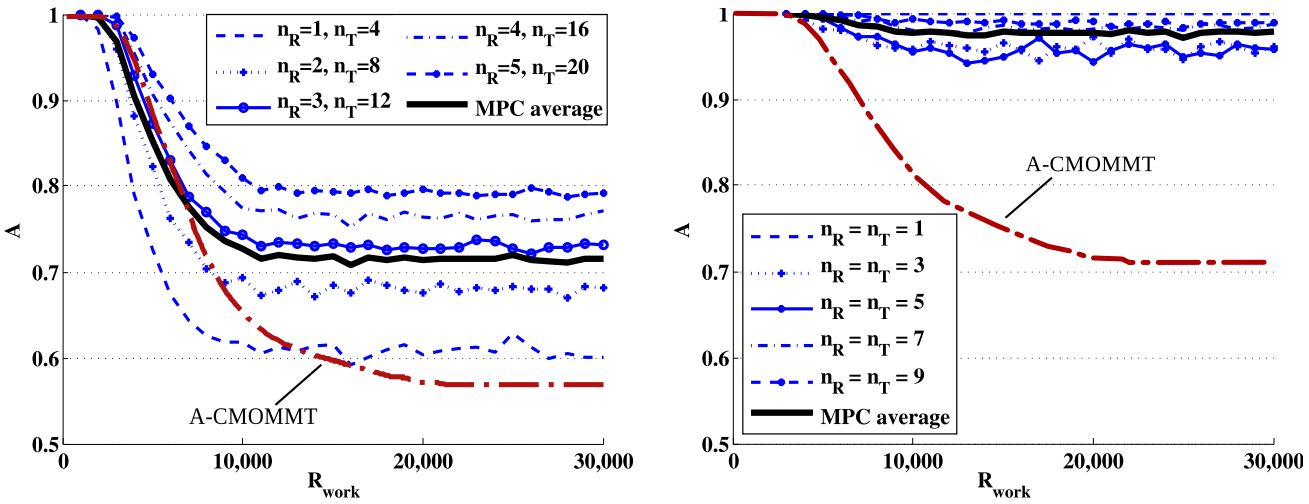
Core CPU, 2.53 GHz, 4GB RAM using the Multi-Parametric Toolbox [80] for Matlab and ILOG CPLEX 11.0 [67] as MILP solver.

From the above results, the values $\widetilde{n}_{Rmax} = \widetilde{n}_{Tmax} = 3$ and $N = 5$ were identified as a suitable tradeoff between computational efficiency and cooperative performance. Using this parameterization, the decentralized MPC approach for the target observation problem was evaluated in simulation for different numbers n_R of robots and n_T of targets in the overall system as well as for different work area radii R_{work}. For each instance (n_R, n_T, R_{work}), 250 simulation runs were performed and evaluated based on the average value of the A-metric.

At the beginning of each simulation run, robots and targets are randomly positioned within a 1000×1000 square in the work area center. Thus, all targets are observed when the simulation starts. Targets are assigned a random orientation and a random velocity up to 150 units per second, which they keep constant during the run. At a 5% chance, they randomly change their orientation (max. $\pm 90°$) at each time step. If a target gets close to the work area boundary, it is repelled and moves on along the reflected direction. Robots can move with a velocity up to 200 units per second.

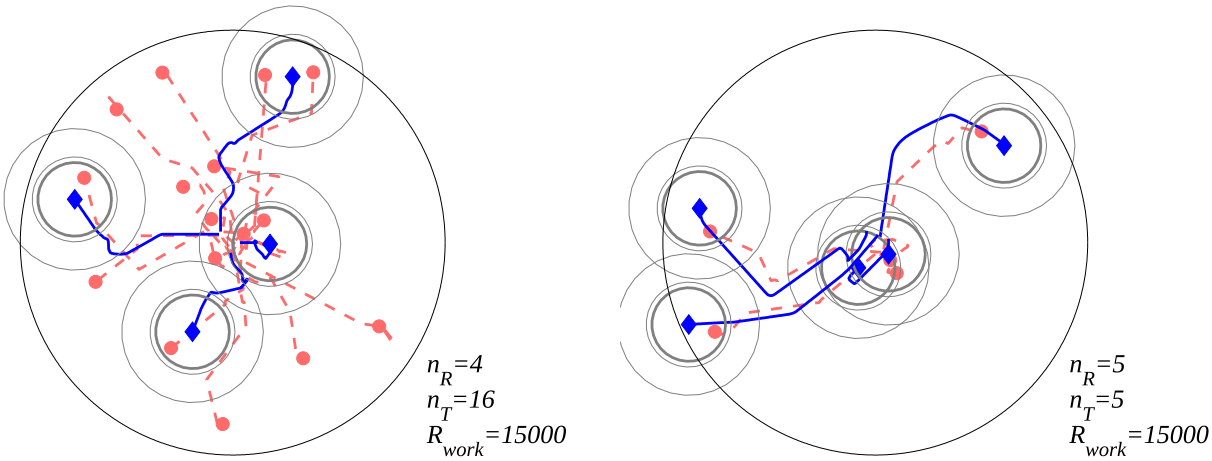
Figure 2.9 shows the results in comparison to those obtained by Parker's heuristic A-CMOMMT approach [105]. Due to the small work area and frequently reflecting targets, an average value of $A = 1$ is obtained for very small values of R_{work}. A then decreases until the work area radius does not influence the success of the observation task anymore. For a robot-target ratio of $\frac{1}{4}$, this is the case for $R_{work} \approx 11000$ for the MPC approach and $R_{work} \approx 22000$ for the A-CMOMMT approach. For $R_{work} > 22000$, the MPC method outperforms the A-CMOMMT method by approx. 25% (cf. Figure 2.9a).

Superiority of the MPC approach becomes even more obvious for a robot-target ratio of 1. Average values for A around 0.98 are obtained for $R_{work} > 10000$ and an improvement of approx. 38% compared to Parker's method is achieved for $R_{work} > 22000$ (see Figure 2.9b). Figure 2.10 depicts examples of simulation runs for the different robot-target ratios.



(a) (b)

Figure 2.9: Comparison of the simulation results obtained from the decentralized MPC approach and from Parker's A-CMOMMT approach [105] based on the average A-metric for a robot-target ratio of $\frac{1}{4}$ (a) and 1 (b) in a work area with radius $R_{work} = 1000, \ldots, 30000$. The larger A, the better the cooperative performance.



(a) Robot-target ratio of $\frac{1}{4}$. **(b)** Robot-target ratio of 1.

Figure 2.10: Examples of simulation runs. Robots are represented by diamond-shaped markers surrounded by circles indicating their observation, sensing, and communication range, respectively. The solid lines represent the robots' trajectories, the dashed lines and circular markers the targets.

2.5.2 Cooperative N-Boundary Tracking

This section summarizes own work published in [46] in cooperation with Andreas Horn and Dominik Haumann.

The application presented in the following deals with the tracking of concentration levels of an atmospheric plume, which is a typical large-scale mission in environmental monitoring. Monitoring in large-scale environments poses additional challenges compared to small or medium-scale environments (cf. [44]) and cooperation of multiple robots for data-gathering offers obvious benefits in these settings.

In order to exploit those, an adaptive sampling strategy for efficient simultaneous tracking of multiple concentration levels of an atmospheric plume by a team of cooperating sensor-equipped UAVs (see Figure 2.11) was developed. Tracking multiple concentration levels by one UAV each efficiently provides not only an accurate estimate of the plume's perimeter but additional information on the concentration gradient.

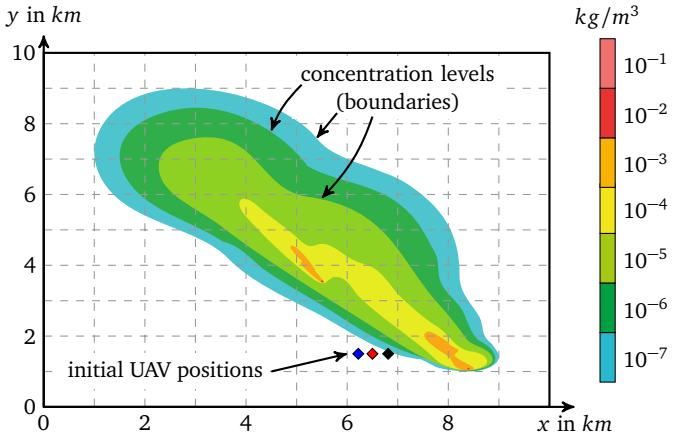


Figure 2.11: Concentration levels of the dispersion of an airborne contaminant. The plume is generated by SCIPUFF [128].

Problem Statement

An airborne contaminant, disseminated in a work area of $10\,\text{km} \times 10\,\text{km} = 100\,\text{km}^2$, is to be estimated by a team of $n_V = 3$ UAVs. Initially, the UAVs are deployed at positions with a low contaminant concentration. The adaptive generation of sampling points for the UAVs combines multiple information sources. It is based on the uncertainty associated with the concentration at the respective location as well as concentration estimates determined from already gathered measurement data.

Since the efficiency of the proposed sampling strategy depends on constant information exchange among the UAVs, stable communication has to be ensured. In order to maintain connectivity and to avoid collisions, decentralized mixed-integer MPC is employed for cooperatively guiding the UAVs to their individual sampling locations, while ensuring that they stay within a maximum distance of $d_{\text{com}} = 3000\,\text{m}$ to each other, but do not get closer than the minimum safety distance $d_{\text{col}} = 20\,\text{m}$.

Objective

The vehicles' task is to cooperatively visit the adaptively generated, discrete measurement locations at a fixed altitude above ground. The goal is to track the concentration levels $l_{c,1} = 10^{-7}\,\frac{\mathrm{kg}}{\mathrm{m}^3}$, $l_{c,2} = 10^{-6}\,\frac{\mathrm{kg}}{\mathrm{m}^3}$ and $l_{c,3} = 10^{-5}\,\frac{\mathrm{kg}}{\mathrm{m}^3}$, where each level is assigned to one specific UAV. This setup is depicted in Figure 2.11.

Modeling

Only selected control-specific details of the approach will be given. For details on the employed sampling strategy, see [46].

UAVs are modeled as point masses with double integrator dynamics employing exact time discretization, i.e. their motion dynamics model corresponds to a 2D version of (2.26). Their maximum velocity is $|v_{\max}| = 10\,\frac{\mathrm{m}}{\mathrm{s}}$ and maximum acceleration $|u_{\max}| = 3\,\frac{\mathrm{m}}{\mathrm{s}^2}$.

A linear approximation of distances d_{vs}^k between a vehicle v and a sampling point (x_s, y_s) is obtained as described in Section 2.4.2. In the same manner, also distances between two UAVs v_i and v_j are approximated. Since the UAVs are required to stay within reach of communication, their distance to each other is limited to a maximum value of d_{com}, and a binary variable $b_{\mathrm{com},ij}^{k\gamma}$ indicates whether this condition holds at time step $k \in \{0,\dots,N\}$:

$$b_{\mathrm{com},ij}^{k\gamma} = 0 \quad \Leftrightarrow \quad (x_{v_i}^k - x_{v_j}^k)\sin\frac{2\pi\gamma}{n_\gamma} + (y_{v_i}^k - y_{v_j}^k)\cos\frac{2\pi\gamma}{n_\gamma} \le d_{\mathrm{com}} \tag{2.39}$$

for $v_i, v_j \in \{1,\dots,\widetilde{n}_V\}, v_i \neq v_j, \gamma = 1,\dots,n_\gamma$. If all variables $b_{\mathrm{com},ij}^{k\gamma} = 0$, the above constraints represent a fully connected communication topology among the vehicles.

Introducing a constraint similar to (2.39) for a minimum distance d_{col} and additional sets of binary variables $b_{\mathrm{col},ij}^{k\gamma}$ and $b_{\mathrm{col},ij}^{k}$ assures that the vehicles do not collide:

$$b_{\mathrm{col},ij}^{k\gamma} = 1 \quad \Leftrightarrow \quad (x_{v_i}^k - x_{v_j}^k)\sin\frac{2\pi\gamma}{n_\gamma} + (y_{v_i}^k - y_{v_j}^k)\cos\frac{2\pi\gamma}{n_\gamma} \le d_{\mathrm{col}} \quad \text{and} \tag{2.40}$$

$$b_{\mathrm{col},ij}^{k} = 1 \quad \Leftrightarrow \quad n_\gamma - \sum_{\gamma=1}^{n_\gamma} b_{\mathrm{col},ij}^{k\gamma} \;\le\; 0\,. \tag{2.41}$$

If $n_\gamma - \sum_{\gamma=1}^{n_\gamma} b_{\mathrm{col},ij}^{k\gamma} = 0$, then v_i and v_j are closer than d_{col} in every direction $\frac{2\pi\gamma}{n_\gamma}$, which should be prevented. This could as well be ensured by a hard constraint $n_\gamma - \sum_{\gamma=1}^{n_\gamma} b_{\mathrm{col},ij}^{k\gamma} \ge 1$. Instead, the indicator variables $b_{\mathrm{col},ij}^{k}$ are used here in order to be able to penalize the violation of inequality (2.41) via the objective function. That way, (2.41) becomes a soft constraint.

Equations (2.39), (2.40), and (2.41) are transformed into mixed-integer linear expressions employing the Big-M method introduced in Section 2.3.2.

The controller's essential purpose is to lead each UAVs to its assigned target location (x_s, y_s), represented as minimization of the distances d_{vs}^k. At the same time, the distance limits induced by the communication and collision constraints are to be met at all times, ensured by penalizing the $b_{\mathrm{com},ij}^{k\gamma}$ and $b_{\mathrm{col},ij}^{k}$ whenever they take value 1. In addition, the UAVs are to move at a

minimum control effort. In summary, the cost function of the CFTOC problem (2.19) takes the following form:

$$\min_{U_N} \sum_{k=0}^{N-1} \Big(q_z \sum_{\nu,s=1}^{\widetilde{n}_V} d_{\nu s}^k + q_\delta \sum_{i=1}^{\widetilde{n}_V - 1} \sum_{j=i}^{\widetilde{n}_V} \big(b_{\mathrm{col},ij}^k + \sum_{\gamma=1}^{n_\gamma} b_{\mathrm{com},ij}^{k\gamma} \big) + q_u |\mathbf{u}^k| \Big) , \tag{2.42}$$

where $\mathbf{u}^k$ concatenates the UAV control inputs $\mathbf{u}_\nu^k$ and $q_z = 10$, $q_\delta = 6000$, $q_u = 0.1$ weight the different objectives.

Results

Simulation results are depicted in Figure 2.12. For clarity, only every 5th sampling point is plotted.

Figure 2.12a shows that the UAVs quickly locate their assigned concentration levels and subsequently track the respective boundaries. At a simulation time of 4h (cf. Figure 2.12b), the UAVs are still close to each other due to the communication constraint. At location (1 km, 6 km), the generated sampling points deviate from the correct concentration level $l_{c,1} = 10^{-7}\,\frac{\mathrm{kg}}{\mathrm{m}^3}$ because of the rather small concentration gradient. However, the outermost vehicle successfully relocates its assigned boundary and the cooperative tracking continues.

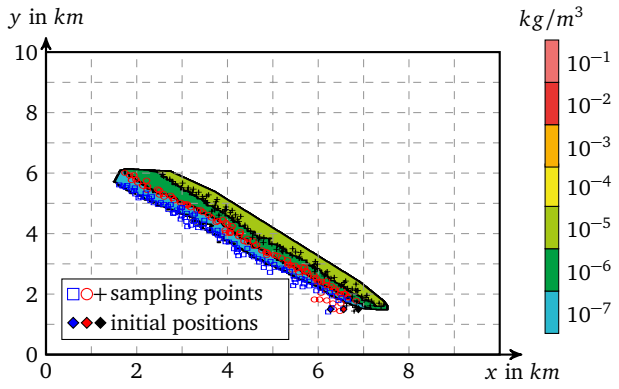
After $T_{\mathrm{sim}} = 6.5\,\mathrm{h}$, the sampling process is completed as depicted in Figure 2.12c. All vehicles succeeded in tracking their individual boundaries. As can be seen in Figure 2.12d, all assigned boundaries have been reconstructed precisely. The motion constraints ensured reliable communication and collision avoidance throughout the entire simulation.

2.5.3 Cooperative Sensing for Process State Estimation

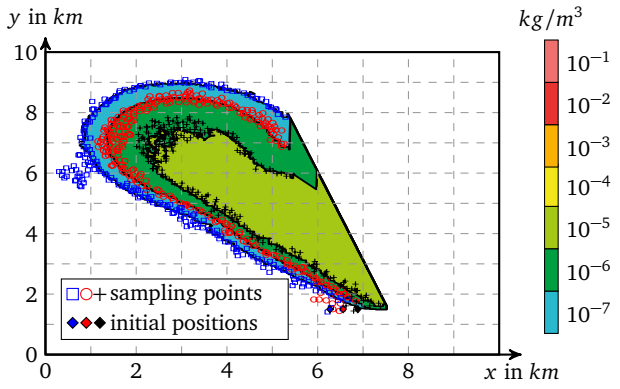
This section summarizes own work published in [126, 47, 127] in cooperation with Tobias Ritter.

The mixed-integer MPC approach has also been successfully combined with a data-driven sampling strategy for online state estimation of atmospheric dispersion processes. The intention was to combine the accuracy of Partial Differential Equation (PDE)-based process models with the potential of a team of optimally cooperating mobile sensors. However, state estimation and vehicle control are considered separate problems that are linked in a repeating sequential procedure. This results in a significant gain of computational efficiency compared to solving a complex optimal control problem incorporating both aspects.

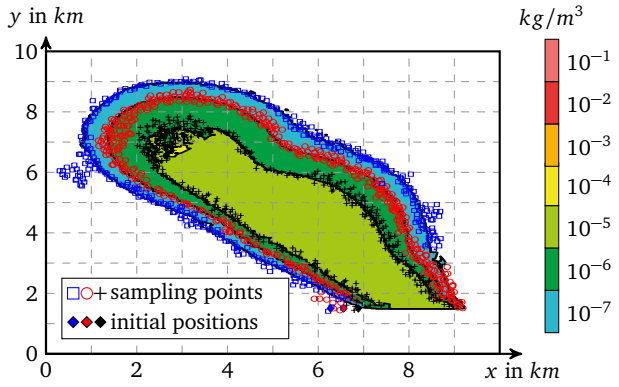
The proposed approach applies an (Ensemble Transform) Kalman Filter (ETKF/KF) for state and uncertainty estimation. Based on the state estimate's error covariance matrix, locations with maximum uncertainty are chosen as future measurement points. Those points are handed to the cooperative controller, which ensures that the sensor vehicles approach the targets in an optimal manner. After assimilating the gathered data with the predicted model state, new measurement locations are determined depending on the uncertainty in the updated state estimate. Repeating this procedure iteratively improves the quality of the state estimate. The required measurements are obtained at optimal exploitation of the vehicles' cooperation and their physical capabilities. A schematic view of this dynamic data-driven concept, first presented in [126], is shown in Figure 2.13.



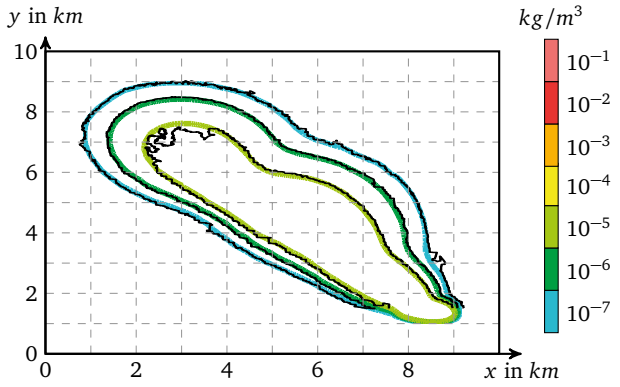
(a) Sampling points and reconstructed concentration levels after 1.5h.



(b) Sampling points and reconstructed concentration levels after 4h.



(c) Sampling points and reconstructed concentration levels after 6.5h.



(d) Comparison of reconstructed and original concentration levels.

Figure 2.12: Simulation results for the N-boundary tracking mission with 3 UAVs.

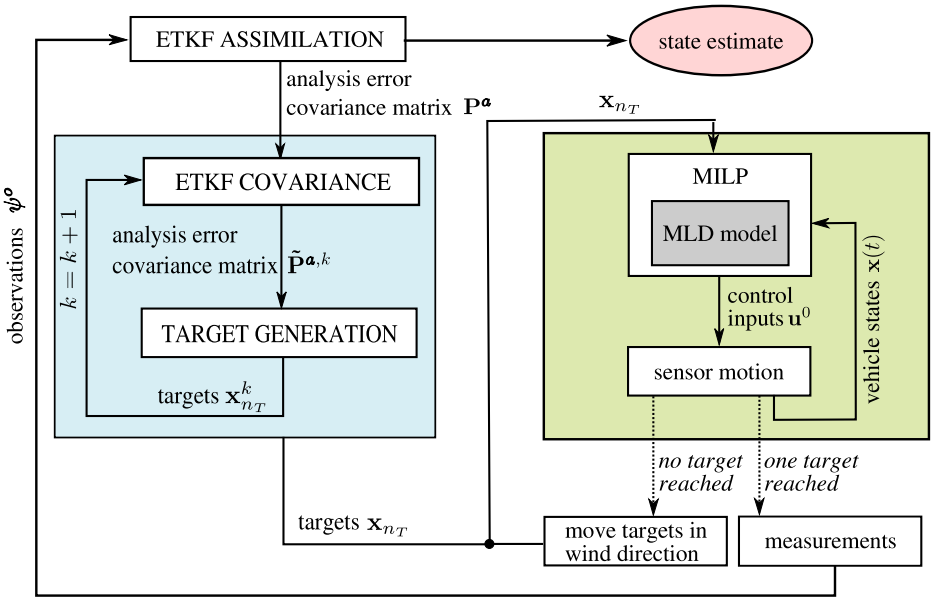


Figure 2.13: Schematic view of the adaptive observation strategy combining target generation based on the Ensemble Transform Kalman Filter (ETKF) and model-predictive cooperative vehicle control. $\mathbf{x}_{n_T}$ comprises the coordinates of all $n_T = n_V$ targets.

In this first version of the approach, the sensors measured only at the target locations. This was modified in [47], where continuous measuring was assumed, attaching more importance to the vehicles' trajectories. Attraction points drawing the vehicles to regions afflicted with high uncertainty were added for this reason. In both [126] and [47], the data-driven observation strategy is implemented in a centralized manner, hence, also a centralized version of the mixed-integer MPC is employed. Using reduced order models further enhanced the PDE-based state estimation and data assimilation method, finally permitting a decentralized application with the decentralized mixed-integer MPC variant as proposed in [127].

Despite the continuous advancement of the data-driven monitoring approach, all three versions share the same primary intention as well as a similar modeling of the sensor vehicles' cooperation, which will briefly be described in the following.

Problem Statement

Sensor-equipped vehicles with a maximum velocity and acceleration of $v_{max} = u_{max} = 0.02$ are deployed in a 4×2 domain in order to collect data on a dynamic atmospheric dispersion process described by an advection-diffusion PDE. For maximized efficiency of the sensing process, valuable measurement locations (referred to as targets) are adaptively determined based on the error covariance matrix provided by a variant of the Kalman Filter, which is used for state estimation of the process as well as for data assimilation (for details on the target generation see [126, 47, 127]). The number of generated targets n_T corresponds to the number of vehicles n_V. Until visited by a vehicle, targets move according to the underlying uniform wind velocity of 0.005 in x-direction. Certain target expiration and recalculation rules ensure the usefulness of local measurements. In order to avoid collisions, every two vehicles are to maintain a minimum distance of $d_{\text{col}} = 0.1$ to each other.

Objective

The vehicles' task is to cooperatively visit the repeatedly updated and slowly moving target locations while avoiding collisions. For the strategy proposed in [47], their trajectories should, in addition, closely bypass vehicle-specific local attraction points. Those are generated at every time step similar to global targets but restricted to a bounded area in the vehicle's direction of motion (see Figure 2.14).

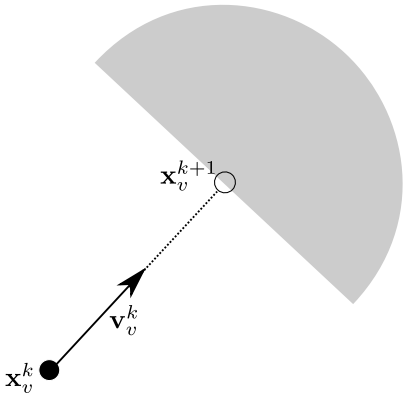


Figure 2.14: Local region (gray) for the selection of attraction points based on a projection of the vehicle's position $\mathbf{x}_v^k$ along its current velocity vector $\mathbf{v}_v^k$.

Modeling

Sensor vehicles are modeled as point mass with double integrator dynamics employing exact time discretization, i.e. their motion dynamics model corresponds to a 2D version of (2.26).

Target movement is predicted according to a linearized representation of the advection influence

$$\mathbf{x}_t^{k+1} = \mathbf{A}_t \mathbf{x}_t^k + \mathbf{b}_t \ , \tag{2.43}$$

where $\mathbf{x}_t^k$, $t \in \{1, \dots, n_T\}$, is the position of target t at time $k \in \{0, \dots, N\}$.

Distances d_{vt}^k between a vehicle v and a measurement target t are approximated as described in Section 2.4.2.

The logical rules expressing whether a measurement point is visited by a vehicle as well as the distances to unvisited targets h_{vt}^k are modeled analog to (2.35), (2.36), and (2.37).

In contrast to Section 2.5.2, collision avoidance between two vehicles v_i and v_j, $i \neq j$ is here modeled by

$$(x_{v_i}^k - x_{v_j}^k) \sin \frac{2\pi\gamma}{n_\gamma} + (y_{v_i}^k - y_{v_j}^k) \cos \frac{2\pi\gamma}{n_\gamma} > d_{min} \quad \Rightarrow \quad b_{\text{col},ij}^{k\gamma} = 0 \tag{2.44}$$

$$\text{and the hard constraint} \quad \sum_{\gamma=1}^{n_\gamma} b_{\text{col},ij}^{k\gamma} \leq n_\gamma - 1 \ . \tag{2.45}$$

As before, all mixed-logical rules are transformed into mixed-integer linear expressions employing the Big-M method introduced in Section 2.3.2.

The objective function for the CFTOC problem (2.19) comprises the minimization of distances h_{vt}^k and of the binary variables s_t^k for cooperatively guiding the vehicles to the target locations. In addition, the control effort is minimized, yielding the combined cost function

$$\min_{U_N} \sum_{k=0}^{N-1} \Big(q_z \sum_{v=1}^{n_V} \sum_{t=1}^{n_T} h_{vt}^k \ + \ q_\delta \sum_{t=1}^{n_T} s_t^k \ + \ q_u |\mathbf{u}^k| \Big) \ , \tag{2.46}$$

where $q_z, q_\delta, q_u \in \mathbb{R}$ weight the different objectives and $\mathbf{u}^k$ concatenates the vehicles' control inputs. In the case of [47], also the controlled vehicle's distance to its individual current attraction point is minimized.

Results

The results depicted in Figures 2.15 and 2.16 have been obtained with centralized MPC using a prediction horizon of $N = 15$ time steps and $\Delta t = 2$. For comparison, two other sensor configurations were applied in the same problem setup. The first consists of three mobile sensors moving randomly, whereas the second employs 16 static sensors evenly distributed in a square area from (-0.6,-0.6) to (0.6,0.6). In both configurations, each sensor takes a measurement every time step, whereas in the proposed adaptive sensing approach, measurements are only taken as soon as one of the vehicles reaches a target location. The test scenario was run 50 times for 120 time steps, each time with a different randomly chosen initial true process state.

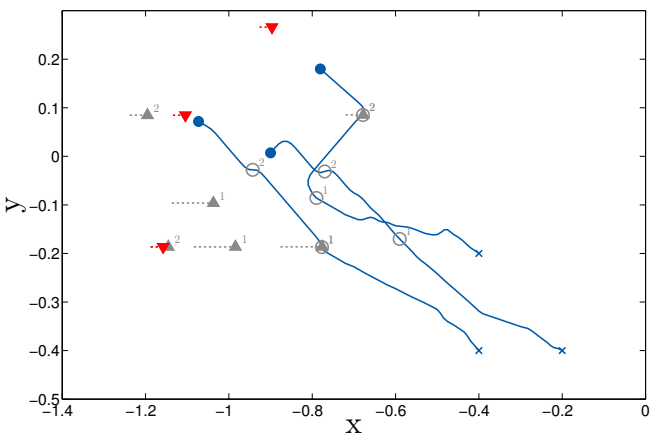


Figure 2.15: Sensors (blue •) and their trajectories with the performed measurements (gray ∘) during an example simulation. Active targets (red ▼) and past targets (gray ▲) are depicted with their wind-induced trajectories. Numbers indicate the generation that targets and measurements belonged to.

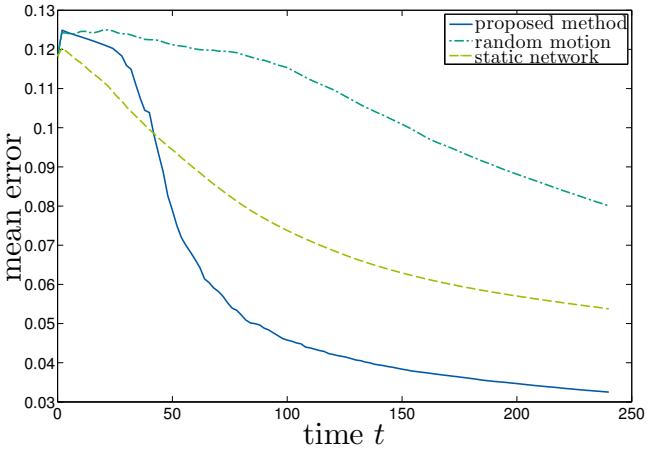


Figure 2.16: Comparison of the mean estimation error over time for three different sensor movement strategies.

Figure 2.16 shows the evolution of the mean error between true and estimated process state over time for the three different observation strategies. Although the proposed strategy results in only ~ 75 measurements within the given time frame – five times less than for the random motion and even 26 times less than for the static sensors – it provides the lowest error.
As can be seen from the example trajectories in Figure 2.15, the vehicles successfully cooperate in visiting the targets and avoid collisions.

Now assuming that the sensors measure at every time step, the influence of local attraction points on the vehicle trajectories was evaluated in terms of a simplified static test scenario depicted in Figure 2.17. The plotted contours represent the uncertainty in the work domain. Centralized MPC with a prediction horizon of $N = 20$ time steps and $\Delta t = 2$ was used.

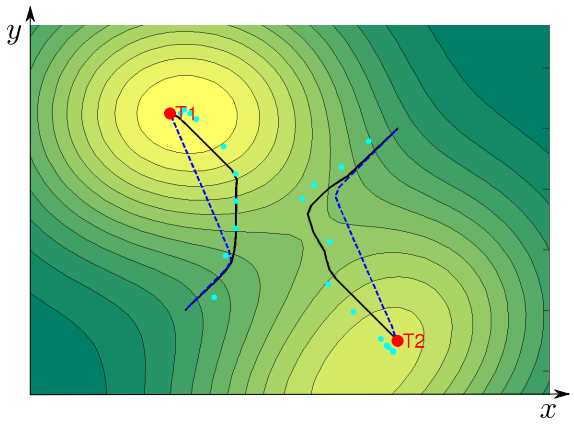
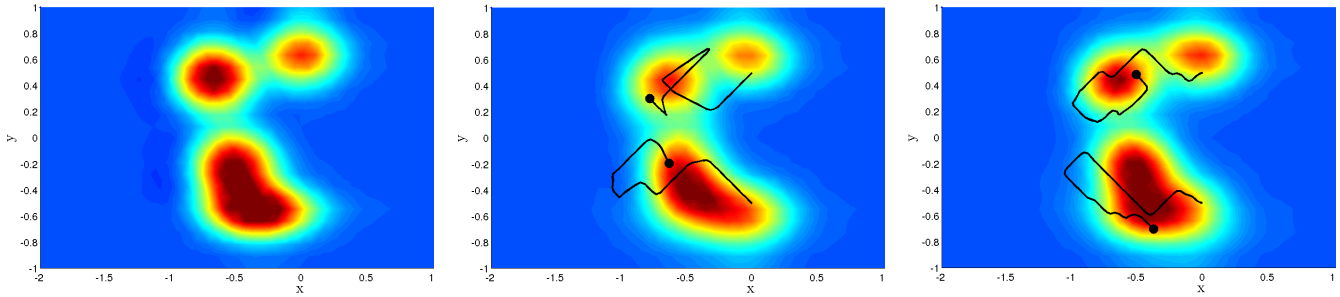


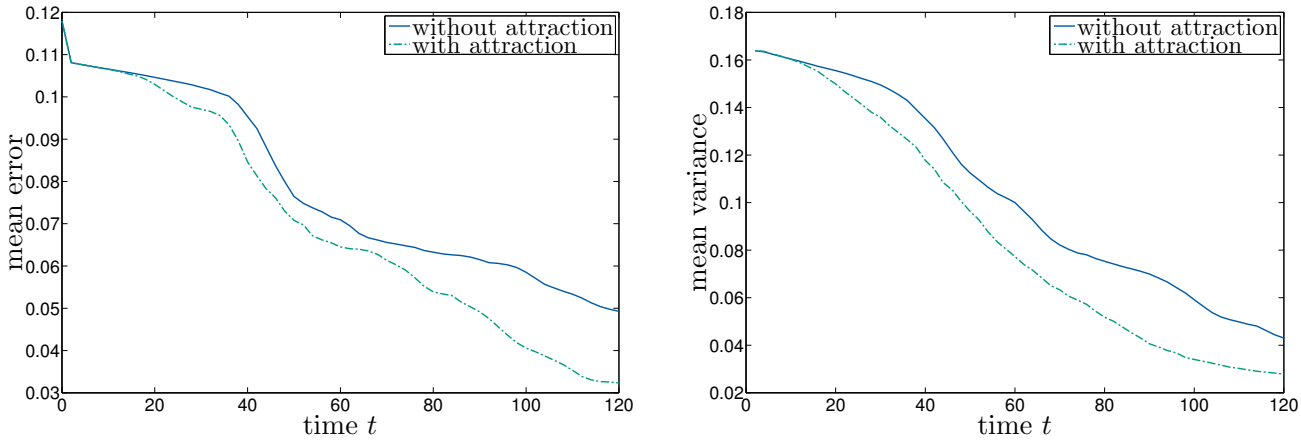
Figure 2.17: Unattracted trajectory (blue dashed line) vs. trajectory (black solid line) influenced by attractor points (light blue, every third point is shown).

The trajectories show that the sensor vehicles first try to minimize the distance to both target points until they head for one target each. The local attraction points influence the trajectories as the sensors are pulled towards locations with higher uncertainty values. This effect can also be observed in the dynamic test case, example results of which are shown in Figure 2.18 comparing sensor trajectories with and without local attraction. How the use of attraction points influences the strategy's performance in terms of estimation error and variance reduction is illustrated in Figure 2.19.



(a) True solution **(b)** Estimate obtained without attraction points **(c)** Estimate obtained with attraction points

Figure 2.18: True and estimated concentration distribution at $t = 120$ with trajectories obtained from the adaptive observation strategy **(b)** without and **(c)** with attraction points. While in **(b)** the vehicles aim at permanently minimizing the distance to both target points, they are deviated to regions of higher uncertainty in **(c)**.



(a) Mean error **(b)** Mean variance

Figure 2.19: Mean estimation error and variance of all grid points over time for the scenario depicted in Figure 2.18, demonstrating the effectiveness of local attraction points.

Figure 2.20 shows an example state estimation result of the enhanced decentralized observation strategy based on reduced order models and decentralized MPC. The performance of the decentralized approach in terms of process state and source function estimation compared to its centralized counterpart as well as a line patrol strategy employing three sensor vehicles is shown in Figure 2.21.

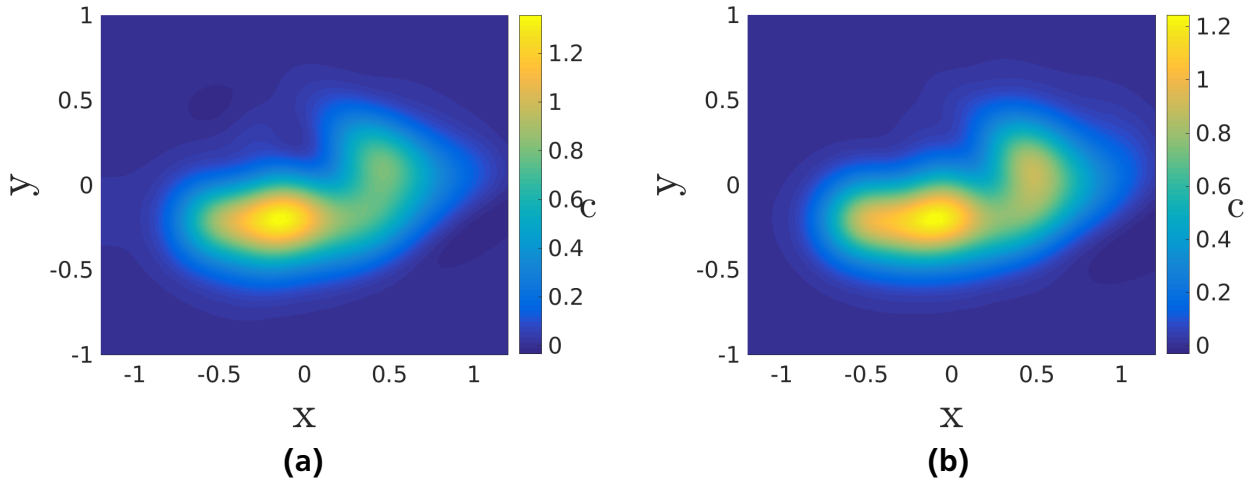
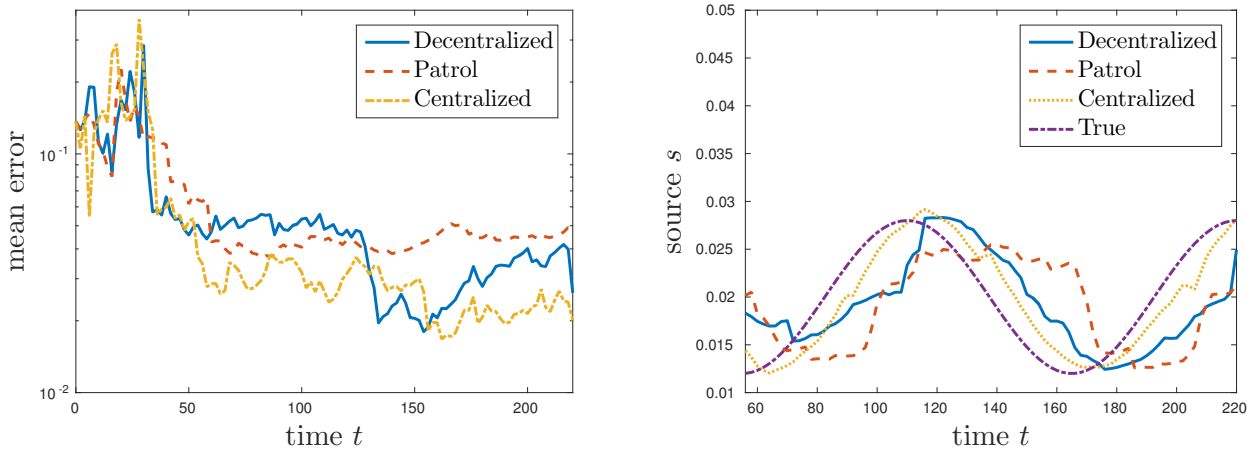


Figure 2.20: Average estimated **(a)** and true **(b)** concentration field after 110 steps of the decentralized observation strategy employing three sensor vehicles.



(a) Error in relation to the true process state over time.

(b) True and estimated source function at the true source location over time.

Figure 2.21: The decentralized adaptive observation strategy in comparison to its centralized version and a line patrol approach.

2.6 Summary

The well-established concept of mixed-integer Model-Predictive Control (MPC) (Section 2.3) has been employed for the design of a novel decentralized optimization-based feedback control approach for multi-vehicle systems faced with cooperative mobility tasks. This class of problems is characterized by high complexity due to the inseparable coupling of continuous and discrete system variables.

Approaches for the abstraction of the vehicles' motion dynamics (Section 2.4.1) and Euclidean distances (Section 2.4.2) have been applied in order to obtain a Mixed Logical Dynamical (MLD) system description. MPC based on MLD systems offers attractive benefits like robustness, stability, and computational efficiency. In addition, the controller's performance can be tuned by various parameters, since its computational efficiency increases with decreasing complexity (i.e. accuracy) of the system description, which has to be traded off based on the considered mission objective. Also the proposed decentralization scheme (Section 2.4.3) can be parameterized to match the available computational resources. The results presented in Section 2.5 confirm the applicability of the proposed control approach in diverse cooperative multi-vehicle monitoring missions and its potential to outperform alternative heuristic cooperative control approaches.

Thus, a flexible, efficient and versatile optimization-based cooperative controller is obtained as the first component of the overall dynamic data-driven cooperative sensing and control approach proposed in this thesis for multiple sensor-equipped vehicles in multi-objective monitoring scenarios. It covers the cooperative control aspect of the considered problem class motivated in Chapter 1. The aspect of adaptive sensing for optimal process estimation is the subject of the following chapter before a combined solution will be proposed in Chapter 4.

3 Dynamic Data-Driven Sensing: A Sequential Optimum Design Approach

This chapter first provides a review of the research questions and related work in the field of adaptive and data-driven sensing for dispersion process monitoring (Section 3.1) before briefly explaining how the new proposed sequential optimum design approach fits in this context (Section 3.2). Section 3.3 then summarizes the theoretical background on atmospheric dispersion modeling, least squares estimation, and Optimum Experimental Design (OED) as required to derive the proposed procedure detailed in Section 3.4. The results of a simulation-based analysis and validation are presented in Section 3.5.

3.1 Related Work in Adaptive Mobile Sensing

Research on environmental robotics has gained a lot of attention in the last decades, primarily starting with maritime robotic systems soon followed by aerial and terrestrial robots [44]. They have been applied in a variety of applications that all have the common aspect of sensor-based information gathering in order to gain insight into the state of certain environmental variables.

This section focuses on existing solutions to the problem of dispersion process monitoring, the large-scale and dynamic characteristics of which make adaptive sensing strategies essential for efficient data collection. *Adaptive sensing* (also termed *adaptive sampling* or *active sensing* in the literature) refers to the question of where to take the next measurement(s) in order to best gain knowledge about a feature of interest.

Source Localization and Boundary Tracking

The process dimensions of maritime or atmospheric dispersion typically prohibit complete coverage, which is why one branch of research narrows the problem to the tracking of boundaries or the localization of pollution sources. Source localization methods employ, among others, gradient information (e.g. [69]), stochastic search approaches (e.g. [66]), or biologically-inspired algorithms (e.g. [61, 85]) to adjust a sensor's motion towards the source. While a single sensor may suffice to solve the localization task, boundary tracking is typically performed by multiple sensor-equipped vehicles. This poses the additional challenge of vehicle coordination in order to avoid collisions [65] or maintain a certain formation [154, 140]. Approaches for the identification and tracking of boundaries range from probability theoretic [71, 70], information theoretic [56, 46] and bio-inspired [35] solutions to relatively simple motion patterns, where vehicles change their moving direction whenever the measured concentration value hits a specified threshold [65, 26, 95]. For a more comprehensive overview of current research on source detection and boundary tracking with wireless sensor networks see the recent survey in [133].

Although some of the approaches are able to simultaneously track multiple concentration levels [46] or to provide additional gradient information [154, 98], the knowledge one gains about the overall dispersion process is limited to certain features and can represent its state at a certain point in time only. Any number of sensor vehicles will only be able to capture parts of the process, especially when considering three-dimensional domains. In order to identify the overall concentration distribution and to enable spatial and temporal extrapolation, it is useful to employ a model of the dispersion process, which leads to the class of model-based approaches for the navigation of sensor vehicles considered in the following.

Model-Aided Path Planning

Many different representations of dispersion processes have been employed in the environmental monitoring literature, among them scalar fields (e.g. [136, 118]), Gaussian processes (e.g. [96, 16, 25]), Gaussian plume/puff models (e.g. [33, 34, 135, 24, 96, 49, 48]), and Distributed Parameter Systems (DPS) or PDEs (e.g. [149, 155, 138, 147, 145, 153, 126, 47, 127, 108]). In order for the representation to best fit the actual process, concentration measurements have to be incorporated, which is referred to as *data assimilation*. Therefore, model-aided path planning aims at guiding mobile sensor platforms, such that the locally obtained measurements are most valuable for improving a state or parameter estimate of the process model.

Optimization methods are well-suited to deal with this problem, which is in contrast to the mainly heuristic solutions for source detection and boundary tracking. Typically, researchers seek to maximize some information measure subject to the sensor vehicle dynamics, i.e. solve an Optimal Control Problem (OCP). In many cases, measures from information theory, such as entropy or uncertainty, are used as performance criterion for the optimization (e.g. [88, 96, 27, 31, 30, 153, 135, 62, 126, 47, 127, 108, 109]. As the motion dynamic constraints add complexity to the problem, some approaches employ a simplified representation of the sensing trajectories and search the next best sensor location e.g. based on the vehicles' maximum motion range [33, 34, 135], Voronoi-constrained movement regions [96], a predefined waypoint graph [16], Greedy search on a grid [31], or a potential field [30].

For answering the question where to observe a phenomenon in order to best estimate the parameters of its model representation, a predestined concept is that of Optimum Experimental Design (OED). Based on the ideas in [117], Uciński developed a comprehensive theoretical framework for the application of OED in the context of DPS identification by multiple mobile sensors [149] that has been the starting point for a number of modified or extended approaches.

As proposed in [149], the authors of [147] optimize the D-optimality criterion from OED based on the Fisher Information Matrix in order to determine optimal control inputs for the sensor vehicles. The novelty of their approach is that the resulting OCP allows for heterogeneity in the sensors' measuring accuracy. A similar idea is pursued in [138], but instead of maximizing the accuracy of the dispersion model parameter estimate, the parameters are assumed to be known and the objective is to minimize the effect of sensor noise.

Since solving the OCP formulations integrating both estimation and vehicle control requires high computational effort, other approaches try to reduce the problem complexity in order to gain efficiency. Although the authors of [155] also employ OED based on the Fisher Information Matrix, they replace the motion dynamics constraints by parameterized vehicle trajectories and,

hence, end up with an NLP formulation that can be solved more efficiently than an OCP. Instead of an OED-based OCP, a Lyapunov-based sensor guidance scheme is proposed in [41].

Model-Aided Dynamic Data-Driven Control

All solutions mentioned so far provide either control trajectories or the sensor trajectories themselves in a feedforward manner. That means that the actual measurements collected by the sensors are not used to adapt their motion to the dynamically changing dispersion process. Control schemes that feed back the observed data in order to update the planned sensor trajectories belong to the class of Dynamic Data-Driven Application Systems (DDDAS), which will be introduced in greater detail in Chapter 4. In [145] and [146], real-time efficient interlaced schemes based on OED for DPS are presented. The OCP is solved for a finite time horizon only, the obtained measurements are incorporated to improve the desired parameter estimate, based on which new trajectories are determined and so on. Similarly, in [62], an NLP minimizing an uncertainty measure is solved in a receding horizon fashion for feedback control of multiple fixed-wing UAVs, and instead of OED, a Kalman filter is used for state and parameter estimation.

Another data-driven guidance scheme for state estimation of a PDE-based dispersion model by multiple sensor UAVs uses virtual attractor particles to guide the vehicles and the standard Kalman filter [108] and the reduced Kalman filter [109], respectively, for data assimilation. Except for the vehicle control part, these two approaches are closely related to those proposed in [126, 47, 127] briefly described in Section 2.5.3. In contrast to the particle-based UAV navigation in [108] and [109], here, cooperation among the vehicles is explicitly treated and the coupling of estimation and control is loosened for efficiency reasons. Moreover, among all aforementioned contributions, only [127] has yet proposed a decentralized data-driven feedback control scheme. For more details on this scheme, the reader is referred to [125]. Similar to this thesis, [125] deals with data-driven atmospheric dispersion monitoring but focuses on state estimation and data assimilation for accurate PDE models without explicitly exploiting the sensor vehicles' motion dynamics.

3.2 Contribution

As can be seen from the literature review above, decentralized data-driven closed-loop control of cooperating sensor-equipped vehicles has rarely been investigated. However, decentrality is essential for obtaining a control scheme that is scalable also to large vehicle teams.
Moreover, only a coupled estimation and control problem accounting for the vehicles' motion dynamics can provide solutions that best exploit the system's potential for optimal estimation results. Yet, the approach has to be efficient enough to be performed on board a UAV with limited computational capacity.

The data-driven sensing strategy presented in this chapter is therefore based on a basic Gaussian puff model instead of a more complex PDE representation of the dispersion process to be identified. In a sequential optimum design procedure, sequences of spatiotemporal waypoints are determined, where measurements are most valuable for improving the parameter estimate of the Gaussian puff model. The design problem is constrained by the UAVs' (possibly heterogeneous) motion dynamics and therefore provides vehicle-specific locally optimal

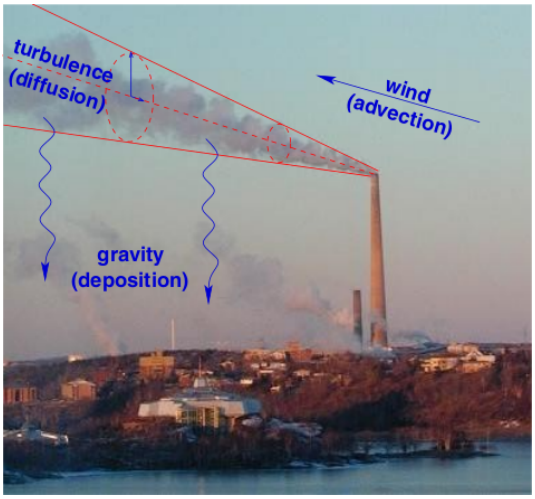


Figure 3.1: Atmospheric contaminant transport is mainly affected by advection (wind), diffusion (turbulent eddy motion), and deposition (gravity). (Source: [139])

waypoint sequences as input for each UAV's motion controller. Parameter estimation, waypoint calculation, and vehicle control are treated separately but permanently interact in a repeating cycle. By coupling the estimation and the control problem in that way, an efficient decentralized data-driven online control scheme is obtained.

3.3 Background

3.3.1 Modeling Atmospheric Dispersion Processes

Advection Diffusion Equation

The term *atmospheric dispersion* denotes the combination of *diffusion* resulting from turbulent eddy motion and *advection* by the wind close to the earth's surface [139]. Figure 3.1 illustrates how both affect the evolution of a plume. Such gas emissions in the atmosphere can be described by the Advection Diffusion Equation (ADE)

$$\frac{\partial c}{\partial t} = -\nabla \cdot \mathbf{q}\,, \tag{3.1}$$

where $c(\mathbf{x}, t)$ is the contaminant concentration in $[\mathrm{kg/m^3}]$ at a point $\mathbf{x} = (x, y, z)$ at time t and $\mathbf{q}$ is the mass flux in $[\mathrm{kg/m^2 s}]$, i.e. the total mass of particles moving through a location within a specified time interval.
$\nabla \cdot \mathbf{q} = \operatorname{div} \mathbf{q} = \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{\partial z}$ denotes the divergence of the vector field $\mathbf{q}$.
Since

$$\mathbf{q} = \mathbf{q}_A + \mathbf{q}_D = c \cdot \mathbf{v}_w - \mathbf{K} \cdot \nabla c\,, \tag{3.2}$$

both diffusion (subscript D) and advection (subscript A) contribute to the total flux $\mathbf{q}$. In this expression, $\mathbf{v}_w = (v_{wx}, v_{wy}, v_{wz})$ is the wind velocity in [m/s] and $\mathbf{K} = \mathrm{diag}(K_x, K_y, K_z)$ is the diffusion coefficient with $K_x(\mathbf{x}), K_y(\mathbf{x}), K_z(\mathbf{x})$ being turbulent eddy diffusivities.

(3.1) is a second-order PDE of parabolic type. It represents the *Eulerian* approach to modeling atmospheric dispersion, i.e. by observing changes in concentration at fixed spatial coordinates. In contrast to that, *Lagrangian* approaches use a non-fixed coordinate system that follows the atmospheric motion. The concentration field is represented by a stream of particles transported by the wind and diffused by turbulences.

(3.1) can either be solved numerically or analytically. Numerical solutions are based on a grid or mesh resulting from a discretization of time and space. Type and size of the mesh determine the approximation accuracy of the solution, but also the required computational effort for finding it, e.g. by employing finite element methods. The numerical approach is to be preferred when dealing with specialized dispersion problems like pollutants spreading over complex terrain where high local precision is desired and computing resources are not limited.

Analytical solutions of the ADE are exact and allow to investigate the influence of different problem parameters more easily [22]. Their derivation is usually based on a number of assumptions that define the solution complexity and, eventually, how close it is to reality. The best known analytical solution to the ADE is the Gaussian model. It is used extensively as the standard approach in literature studying industrial emissions, various pollutant transport processes as well as the release of nuclear or biological contaminants [139]. Environmental agencies all over the world use advanced dispersion simulation software based on Gaussian models for regulatory applications [22], such as the steady-state plume model AERMOD [1] or the non-steady-state puff model CALPUFF [2].

For disaster response scenarios, however, the importance of less complex (Gaussian) dispersion models that offer "a quick, simple, hands-on prediction capability for plume direction, coverage, and lethality" for first responders is emphasized in [131]. The Gaussian puff model employed in this thesis fulfills these requirements and will be introduced in the following section. However, the optimum design approach proposed in Section 3.4 could equally well deal with other closed-form analytical solutions of (3.1). It is possible to extend the Gaussian puff or plume solutions to represent more complex dispersion phenomena, e.g. multiple sources or stronger turbulences. In general, more complex solutions of (3.1) can be derived by changing the underlying assumptions and boundary conditions (cf. [139]).

Gaussian Puff Solution

The derivation of the Gaussian puff solution used in this thesis for modeling atmospheric dispersion processes is based on [73]. What follows is a number of simplifying assumptions and resulting boundary conditions for analytically solving (3.1):

- The contaminant mass Q is instantaneously released from a point source located at (x_0, y_0, z_0) at time t_0:

$$c(x, y, z, t_0) = Q\delta(x - x_0)\delta(y - y_0)\delta(z - z_0)\,, \tag{3.3a}$$

where $\delta(\cdot)$ is the Dirac delta function with

$$\delta(x) = 0 \quad \text{for} \quad x \neq 0 \quad \text{and} \quad \int_{-\infty}^{\infty} \delta(x)dx = 1 \, .$$

- The wind velocity is assumed to be constant and restricted to the x direction, i.e. $\mathbf{v}_w = (v_{wx}, 0, 0) = const,\ v_{wx} \geq 0$.
- K_x, K_y, K_z are constants.
- The topography of the ground is neglected.
- The contaminant concentration converges to zero far from the source:

$$c \to 0 \quad \text{for} \quad x \to \pm\infty, y \to \pm\infty, z \to \infty \, . \tag{3.3b}$$

- The vertical flux vanishes at the ground surface:

$$\frac{\partial c}{\partial z}(x, y, 0, t) = 0 \, . \tag{3.3c}$$

Applying the assumptions above, (3.1) can be stated as

$$\begin{aligned} \frac{\partial c}{\partial t} &= -\nabla \cdot \mathbf{q} \\ &= -\nabla \cdot \left(c v_{wx} - K_x \frac{\partial c}{\partial x}, -K_y \frac{\partial c}{\partial y}, -K_z \frac{\partial c}{\partial z} \right)^T \\ &= -v_{wx} \frac{\partial c}{\partial x} + K_x \frac{\partial c}{\partial x^2} + K_y \frac{\partial c}{\partial y^2} + K_z \frac{\partial c}{\partial z^2} \end{aligned} \tag{3.4}$$

and is to be solved subject to the constraints (3.3).

By Laplace and Fourier transforms [73] or using Green's function [139], one can obtain the *Gaussian puff solution*

$$C(\boldsymbol{\theta}, x, y, z, t) = \frac{Q}{8\pi^{\frac{3}{2}}(K_x K_y K_z)^{\frac{1}{2}} \Delta t^{\frac{3}{2}}} e^{-\frac{(\Delta x - v_{wx}\Delta t)^2}{4K_x \Delta t} - \frac{\Delta y^2}{4Ky\Delta t}} \cdot \left(e^{-\frac{(z-z_0)^2}{4K_z \Delta t}} + e^{-\frac{(z+z_0)^2}{4K_z \Delta t}} \right) , \tag{3.5}$$

where $\Delta x = x - x_0$, $\Delta y = y - y_0$, and $\Delta t = t - t_0$.

Further assuming that the lateral eddy diffusion in x and y direction is identical, i.e. $K_x = K_y$, and that K_z can be derived from the theoretical model $K_z = \beta(z - z_0)^\gamma$ [73] with β and γ depending on the atmospherical conditions, the vector of unknown parameters for (3.5) is $\boldsymbol{\theta} = (Q, K_x, x_0, y_0, z_0, t_0)^T$. Figure 3.2 shows an example snapshot of a Gaussian puff 20 min after the instantaneous contaminant release.

3.3.2 Nonlinear Least Squares

In order to estimate the parameters of the Gaussian puff solution based on measurements gathered by the UAVs' onboard sensors, a data fitting (or parameter estimation) problem is set up.

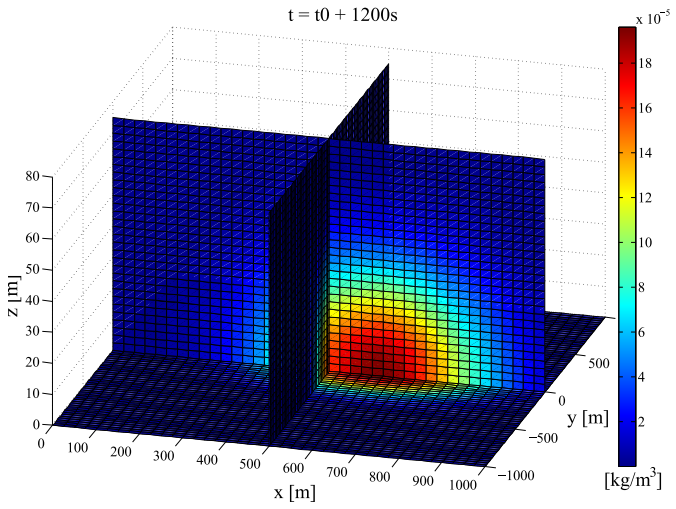


Figure 3.2: Three-dimensional plot of the concentration distribution according to the Gaussian puff solution (3.5) 1200 seconds after the gas release for $\boldsymbol{\theta} = (1000\,\mathrm{kg}, 12\,\mathrm{m}^2/\mathrm{s}, 2\,\mathrm{m}, 5\,\mathrm{m}, 0\,\mathrm{m}, 2\,\mathrm{s})$.

Due to the nonlinearity of (3.5), this problem is nonlinear and, in addition, has to account for measurement errors due to sensor noise as will be stated more formally in the following.

Let $(\mathbf{p}_i, v_i)$, $i = 1, 2, \ldots, m$, be m data points with $\mathbf{p}_i = (x_i, y_i, z_i, t_i)$ containing the coordinates in space and time of the ith measurement (or observation) of a feature of interest yielding the value $v_i \in \mathbb{R}$ defined as

$$v_i = \mathcal{M}(\boldsymbol{\mu}, \mathbf{p}_i) + \epsilon_i, \qquad i = 1, 2, \ldots, m\,, \tag{3.6}$$

where $\mathcal{M}(\boldsymbol{\mu}, \mathbf{p}_i)$ is a general model of the true response of the process to be estimated, which depends on the (unknown) parameters $\boldsymbol{\mu} \in \mathbb{R}^{n_\mu}$. The measurement error ϵ_i is assumed to be white Gaussian noise with mean $\mathrm{E}(\epsilon_i) = 0$ and variance $\mathrm{var}(\epsilon_i) = \sigma_i^2$, where σ_i is the standard deviation of the error. The general data fitting objective is to identify the parameters $\boldsymbol{\mu}$ of the model $\mathcal{M}(\boldsymbol{\mu}, \mathbf{p})$ such that it best fits the noisy data.

In addition, it is assumed that the measurement errors

(1) are uncorrelated, i.e. $\mathrm{E}(\epsilon_i \epsilon_j) = \mathrm{cov}(\epsilon_i, \epsilon_j) = 0$ for $i \neq j$, and

(2) have identical variance, i.e. $\mathrm{var}(\epsilon_i) = \sigma^2$.

The parameter estimation problem can formally be stated as follows.

Definition 3.3.1. ***Nonlinear least squares problem (NLLSQ)*** **[60].** In the case of white noise (where the errors have a common variance σ^2), find a minimizer $\boldsymbol{\mu}^*$ of the nonlinear objective function f with the special form

$$\min_{\boldsymbol{\mu}} f(\boldsymbol{\mu}) := \min_{\boldsymbol{\mu}} \frac{1}{2} ||\mathbf{r}(\boldsymbol{\mu})||_2^2 = \min_{\boldsymbol{\mu}} \frac{1}{2} \sum_{i=1}^{m} r_i(\boldsymbol{\mu})^2\,, \tag{NLLSQ}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{n_\mu}$ and $\mathbf{r}(\boldsymbol{\mu}) = (r_1(\boldsymbol{\mu}), \ldots, r_m(\boldsymbol{\mu}))^T$ is a vector-valued function of the residuals

$$r_i(\boldsymbol{\mu}) = v_i - \mathcal{M}(\boldsymbol{\mu}, \mathbf{p}_i), \qquad i = 1, 2, \ldots, m \tag{3.7}$$

for each data point $(\mathbf{p}_i, v_i)$ and model function $\mathcal{M}$.

In contrast to linear least squares problems, there is no closed-form solution if the model $\mathcal{M}$ is nonlinear in some or all parameters $\boldsymbol{\mu}$. This is the case if at least one of the parameters in $\boldsymbol{\mu}$ appears nonlinearly, i.e. its derivative $\frac{\partial M}{\partial \mu_i}$ is a function of μ_i [60]. The solution then has to be found iteratively, typically by using either a Gauss-Newton or a Levenberg-Marquardt type algorithm. A maximum likelihood estimator for the unknown parameters $\boldsymbol{\mu}$ is obtained if assumptions (1) and (2) hold [97].

In the more general case, the standard deviations σ_i of the measurement errors are not identical. In this case, the residuals (3.7) have to be weighted by the reciprocals of the standard deviations, which leads to the formulation of the *weighted nonlinear least squares problem*:

$$\min_{\boldsymbol{\mu}} \frac{1}{2}||\mathbf{W}\mathbf{r}(\boldsymbol{\mu})||_2^2 = \min_{\boldsymbol{\mu}} \frac{1}{2}\sum_{i=1}^{m}(w_i r_i(\boldsymbol{\mu}))^2 = \min_{\boldsymbol{\mu}} \frac{1}{2}\sum_{i=1}^{m}\left(\frac{r_i(\boldsymbol{\mu})}{\sigma_i}\right)^2 , \tag{3.8}$$

where $\mathbf{W} = \text{diag}(w_1, w_2, \ldots, w_m) = \text{diag}(\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_m^{-1})$. Within the scope of this thesis, these weights are assumed to be known. However, this is not the general case. In practice, $\mathbf{W}$ has to be estimated.

3.3.3 Optimum Experimental Design for Dynamic Processes

"A well-designed experiment is an efficient method of learning about the world."

This is the first sentence in the monograph on Optimum Experimental Design (OED) by Atkinson et al. [11]. In the context of data-driven monitoring by sensor-carrying UAVs, the term *experiment* means the process of data collection at the most informative locations in both time and space. *Designing* the experiment then refers to computing these locations that are optimal in that the corresponding measurement data minimizes the uncertainty in the parameter estimate obtained from (3.8). The theory of OED provides a powerful mathematical basis for the development of algorithms for design construction. This section gives a firm overview of the theoretical background of the design construction method employed in this thesis, mainly following the notation and descriptions in [11].

The design of experiments for the identification of the parameters in the nonlinear Gaussian puff solution (3.5) is challenging since designs for nonlinear models depend on the values of the parameters $\boldsymbol{\mu}$, hence, they are only locally optimal. In the optimum design literature, this fact is often referred to as the "chicken-and-egg" problem and is nicely captured by a quote given in [149]: *"You tell me the value of $\boldsymbol{\mu}$ and I promise to design the best experiment for estimating $\boldsymbol{\mu}$."* In order to overcome this interdependence and to reduce negative effects of a bad choice of $\boldsymbol{\mu}$ on the design, *sequential designs* or designs for several prior values of $\boldsymbol{\mu}$ can be used to obtain near-optimum solutions.

The general scheme of sequential designs can be summarized as follows (cf. [11]):

1. Make initial guess of parameters.
2. Linearize model by Taylor series expansion about parameter estimate.
3. Find optimum design for linearized model.
4. a) Perform one or a few trials to obtain new observations.

b) Update estimate of $\boldsymbol{\mu}$.

c) Stop if estimate is accurate enough.

d) Otherwise repeat from step 2 for new estimate.

In general, it is most efficient to update the parameter estimate after each new observation. However, depending on the cost of conducting the experiment and for computing the design, in practice, it might be more suitable to collect more than one measurement before recalculating the design. While this might be disadvantageous for the parameter estimation at the beginning of the sequential procedure, the number of observations between recalculations of the estimate can safely be increased as the estimate improves.

The remainder of this section will give some more theoretical insight on finding optimum designs for linear models as required for step 3) of the general sequential design scheme. The procedure as applied in the data-driven monitoring context will then be stated more precisely in Section 3.4.

Information Matrix for Linear Models

Let $\mathcal{M}_{\text{lin}}(\boldsymbol{\mu}, \mathbf{p}_i) = \mathbf{g}(\mathbf{p}_i)^T \boldsymbol{\mu}$ be a general linear process model depending on the n_μ parameters in $\boldsymbol{\mu} \in \mathbb{R}^{n_\mu}$, such that the expected observation values $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_m)^T$ corresponding to the trials at $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m$ can be written as

$$E(\boldsymbol{\nu}) = \mathbf{G} \cdot \boldsymbol{\mu} \,, \tag{3.9}$$

where $\mathbf{g}(\mathbf{p}_i)^T$ forms the ith row of $\mathbf{G}$. An estimator $\bar{\boldsymbol{\mu}}$ of the linear least squares problem

$$\min_{\boldsymbol{\mu}} f_{\text{lin}}(\boldsymbol{\mu}) = \min_{\boldsymbol{\mu}} \frac{1}{2} ||\boldsymbol{\nu} - \mathbf{G}\boldsymbol{\mu}||_2^2 \tag{3.10}$$

would have to satisfy $\nabla f_{\text{lin}}(\bar{\boldsymbol{\mu}}) = 0$ and hence the normal equations

$$\mathbf{G}^T \mathbf{G} \bar{\boldsymbol{\mu}} = \mathbf{G}^T \boldsymbol{\nu} \,. \tag{3.11}$$

The $n_\mu \times n_\mu$ matrix $\mathbf{G}^T\mathbf{G}$ is called the *information matrix* for $\boldsymbol{\mu}$. The larger the entries of $\mathbf{G}^T\mathbf{G}$, the more information is obtained from an experiment with trials at $\mathbf{p}_i$, $i = 1, 2, \ldots, m$.

From a statistical point of view, the reliability of the estimate $\bar{\boldsymbol{\mu}}$ can be expressed in terms of a $(1-\alpha) \cdot 100\%$ confidence ellipsoid: If the m-trial experiment was repeated multiple times, the fraction of confidence ellipsoids (individually calculated for each experiment) that cover the true parameter values would tend to $(1-\alpha)\cdot 100\%$. The smaller the confidence region, the more reliable is the estimate. Since the volume of this ellipsoid is inversely proportional to $\sqrt{|\mathbf{G}^T\mathbf{G}|}$, it is desirable to maximize the determinant $|\mathbf{G}^T\mathbf{G}|$, which leads to so-called *D-optimum designs*. They will, along with other examples of optimality criteria, be introduced more formally on page 53.

In OED theory, a design is represented by a probability measure ξ that defines the distribution of trials over the *design region* Ξ. One distinguishes between *continuous designs*

$$\xi = \begin{Bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n \\ \omega_1 & \omega_2 & \cdots & \omega_n \end{Bmatrix} , \tag{3.12}$$

where n is the number of distinct design points, $\int_\Xi \xi d\mathbf{p} = 1,\ 0 \le \omega_i \le 1$, and $\sum \omega_i = 1$ and *exact designs*

$$\xi_m = \begin{Bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n \\ l_1/m & l_2/m & \cdots & l_n/m \end{Bmatrix} , \tag{3.13}$$

where the total number of trials m is fixed, the number of trials l_i at a specific design point $\mathbf{p}_i$ has to be integer, and $\sum_{i=1}^{n} l_i = m$.

The advantage of working with measures instead of m-trial designs is mainly of theoretical nature. It is more convenient for the theory and construction of designs to work with continuous rather than discrete mathematics. It is obvious, though, that only exact designs can be put into practice, but in general, they do not represent the optimum design. An integer approximation of the optimum continuous design ξ^* can typically yield a good exact design.

While, previously, we derived the information matrix for m-trial designs

$$\mathbf{G}^T\mathbf{G} = \sum_{i=1}^{m} \mathbf{g}(\mathbf{p}_i)\mathbf{g}^T(\mathbf{p}_i) , \tag{3.14}$$

a normalized formulation

$$\mathbf{M}(\xi) = \int_\Xi \mathbf{g}(\mathbf{p})\mathbf{g}^T(\mathbf{p})\xi d\mathbf{p} = \sum_{i=1}^{m} \omega_i \mathbf{g}(\mathbf{p}_i)\mathbf{g}^T(\mathbf{p}_i) \tag{3.15}$$

is obtained for continuous designs ξ and

$$\mathbf{M}(\xi_m) = \frac{\mathbf{G}^T\mathbf{G}}{m} \tag{3.16}$$

for exact designs ξ_m.

Another useful definition for m-trial designs is that of the *predicted response*

$$\bar{\nu}(\mathbf{p}) = \mathbf{g}^T(\mathbf{p})\bar{\boldsymbol{\mu}} \tag{3.17}$$

with variance

$$\mathrm{var}(\bar{\nu}(\mathbf{p})) = \sigma^2 \mathbf{g}^T(\mathbf{p})(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{g}(\mathbf{p}) . \tag{3.18}$$

As before, σ^2 denotes the variance of the measurement error at point $\mathbf{p}$.

For continuous designs, the *standardized variance of the predicted response* is defined by

$$d(\mathbf{p}, \xi) = \mathbf{g}^T(\mathbf{p})\mathbf{M}^{-1}(\xi)\mathbf{g}(\mathbf{p}) \tag{3.19}$$

and by

$$d(\mathbf{p}, \xi_m) = \mathbf{g}^T(\mathbf{p})\mathbf{M}^{-1}(\xi_m)\mathbf{g}(\mathbf{p}) = \frac{m\,\mathrm{var}(\bar{\nu}(\mathbf{p}))}{\sigma^2} \tag{3.20}$$

for exact designs.

Sequential Construction of Exact Designs

For the construction of exact designs with a specified number of trials m, it is common to use a sequential algorithm that starts with an m_0-trial exact design ξ_0 and iteratively adds a trial at the point $\mathbf{p}_i$ with the greatest variance of the predicted response, i.e.

$$d(\mathbf{p}_i, \xi_i) = \max_{\mathbf{p}\in\Xi} d(\mathbf{p}, \xi_i) \quad i = 1, 2, \ldots, m - m_0 \; . \tag{3.21}$$

In each iteration i, the information matrix $\mathbf{M}(\xi_{i+1})$ for the new design is updated according to

$$\mathbf{M}(\xi_{i+1}) = \frac{m}{m+1}\mathbf{M}(\xi_i) + \frac{1}{m+1}\mathbf{g}(\mathbf{p}_i)\mathbf{g}(\mathbf{p}_i)^T \; . \tag{3.22}$$

As $i \to \infty$, the resulting exact design approximates the D-optimum continuous design ξ^*. Hence, the exact design improves with increasing m.

A difficulty that has to be tackled during design construction is that the optimality criteria for exact designs do not yield convex optimization problems and have many local optima. If possible, it is, therefore, advisable to run the design search starting from many different initial designs ξ_0.

Once an m-trial exact design is found, it can be further improved, e.g. by replacing design points by others with greater variance of the predicted response.

Optimality Criteria

The relation between D-optimum designs and the confidence ellipsoid for the unknown parameters $\boldsymbol{\mu}$ has already been mentioned in Section 3.3.3. In the following, it will be reconsidered with regard to the eigenvalues of the information matrix $\mathbf{M}(\xi)$ (as defined by (3.15)) and more examples of optimality criteria will be introduced and illustrated by means of the confidence ellipsoid.

Consider the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_{n_\mu}$ of $\mathbf{M}(\xi)$. If $\mathbf{M}(\xi)$ is nonsingular, then $\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_{n_\mu}^{-1}$ are the eigenvalues of $\mathbf{M}^{-1}(\xi)$. The latter are proportional to the squares of the lengths of the axes of the confidence ellipsoid for $\boldsymbol{\mu} \in \mathbb{R}^{n_\mu}$.

From that, the following optimality criteria can be derived and are illustrated in Figure 3.3:

- **A-optimum** designs minimize $\Psi_A(\mathbf{M}) = \sum_{i=1}^{n_\mu} \frac{1}{\lambda_i}$ or equivalently $\Psi_A(\mathbf{M}) = \mathrm{tr}(\mathbf{M}^{-1}(\xi))$, which corresponds to minimizing the diagonal of the bounding box of the confidence ellipsoid.
- **D-optimum** designs minimize $\Psi_D(\mathbf{M}) = \prod_{i=1}^{n_\mu} \frac{1}{\lambda_i}$ or equivalently $\Psi_D(\mathbf{M}) = \ln|\mathbf{M}^{-1}(\xi)|$, which corresponds to minimizing the volume of the confidence ellipsoid. Taking the logarithm of the determinant yields a convex criterion formulation.
- **E-optimum** designs minimize $\Psi_E(\mathbf{M}) = \max_i \frac{1}{\lambda_i}$ or equivalently $\Psi_E(\mathbf{M}) = \lambda_i(\mathbf{M}(\xi))$, which corresponds to minimizing the largest radius of the confidence ellipsoid.

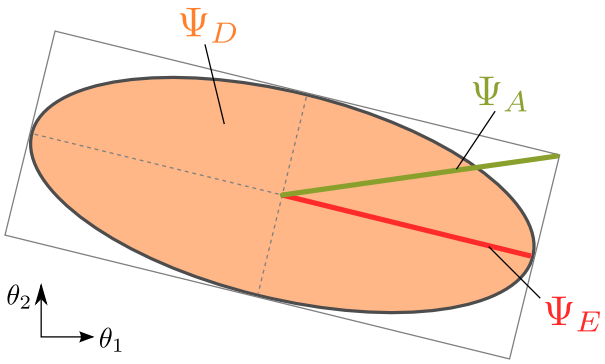


Figure 3.3: Confidence ellipsoid in 2D and its relation to the A-, D-, and E-optimality criterion.

Another important class that is not directly related to the parameters' confidence is that of

- **G-optimum** designs minimizing $\Psi_G = \bar{d}(\xi) = \max_{\mathbf{p}\in\Xi} d(\mathbf{p}, \xi)$, i.e. the maximum standardized variance of the predicted response.

All of the above criteria are convex, so the theory of convex analysis and optimization applies and globally optimal solutions can be found. The mentioned criteria are suitable for the estimation of all parameters in $\boldsymbol{\mu}$ by experimental design. Extended formulations for the estimation of parameter subsets exist along with many more specialized optimality criteria, more detailed descriptions of which can, e.g., be found in [11].

It is also possible to use a linear combination of A-, D-, E-, and G-optimality to formulate a custom optimality criterion. The criteria are all based on minimizing the variance of the parameter estimate and closely related. Therefore, an optimal solution for one criterion is typically near-optimal in terms of the other criteria. As will be seen below, continuous G-optimum designs are even equivalent to continuous D-optimum designs, a fact that can be exploited during design construction.

For choosing an appropriate optimality criterion for a certain model and use case, a performance evaluation of designs obtained with different criteria is useful. For the application considered in this thesis, the performance of A-, D-, and E-optimality has been compared. The results will be presented in Section 3.5.1.

General Equivalence Theorem

The General Equivalence Theorem states that under the mild assumptions that Ξ is compact and the optimality criterion Ψ is convex and differentiable, a design that minimizes Ψ at the same time also optimizes some other optimum design criterion. It is typically formulated for the equivalence of D- and G-optimality, but can analogously be applied to other criteria.

Theorem 3.3.1. ***General Equivalence [144].*** Let ξ^* be an optimum continuous design. Then the following three requirements on ξ^* are equivalent:

(1) ξ^* maximizes $|\mathbf{M}(\xi)|$.

(2) ξ^* minimizes $\bar{d}(\xi)$.

(3) $\bar{d}(\xi^*) = n_\mu$, where n_μ is the number of linearly independent model parameters.

All designs satisfying (1)-(3), as well as their convex combinations, have the same information matrix $\mathbf{M}(\xi^*)$.

The equivalence of two criteria can be useful for the construction, but also for the checking of designs, which are not necessarily unique. For example, a design is D-optimal iff it satisfies one of the other two conditions. It is important to note, though, that above theorem does, in general, not hold for exact designs since for an exact G-optimum design $\xi_{G,m}$ with m trials it is possible that $\bar{d}(\xi_{G,m}) > n_\mu$.

Another interesting characteristic of optimum designs is that the number n of distinct design points is bounded by $n_\mu(n_\mu+1)/2$. This is due to the information matrix $\mathbf{M}(\xi)$ being a symmetric $n_\mu \times n_\mu$ matrix that can be decomposed into a weighted sum of $n_\mu(n_\mu+1)/2$ rank-one information matrices $\mathbf{M}(\bar{\xi}_i)$.

3.4 Proposed Sequential Optimum Design of Vehicle-Specific Sensing Trajectories

Parts of this section have been published in [49].

This section describes how the theory of nonlinear data fitting and OED is applied for computing optimized trajectories for sensor-equipped UAVs in order to best estimate the parameters of a Gaussian puff model of an atmospheric dispersion process. The sensing trajectories are represented as sequences of waypoints, where concentration measurements are most informative for the parameter estimation. These waypoint sequences are tailored to the motion dynamics of each individual vehicle and will later serve as input for each UAV's motion controller. Parameter estimation, waypoint calculation, and vehicle control permanently interact as illustrated in Figure 3.4 and an efficient decentralized adaptive mobile sensing approach is obtained.

After introducing the sensor model in Section 3.4.1, the information matrix as the basis for the optimum design problem will be derived in Section 3.4.2. The constrained design problem and its characteristics are introduced in Section 3.4.3 followed by a description of the overall sequential procedure in Section 3.4.4. The section concludes by proposing two different options for the decentralization of the approach in Section 3.4.5.

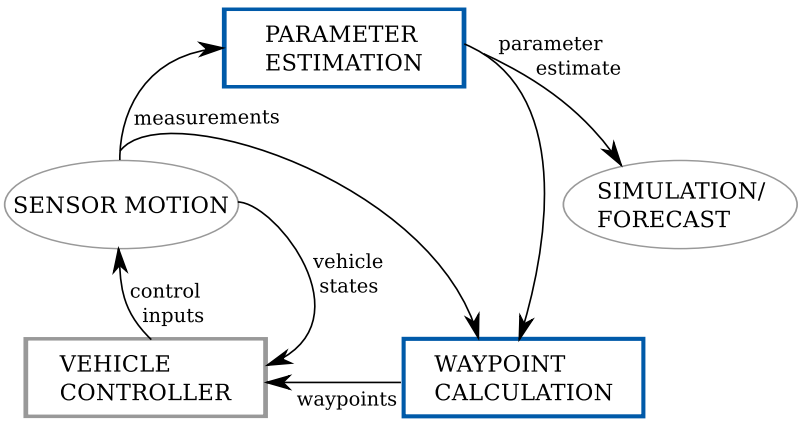


Figure 3.4: Schematic view of the adaptive mobile sensing loop.

3.4.1 Sensor Model and Parameter Estimation

The objective of the parameter estimation component in the adaptive sensing loop illustrated in Figure 3.4 is the identification of the parameters $\boldsymbol{\theta} = (Q, K_x, x_0, y_0, z_0, t_0)^T$ of the Gaussian puff solution (3.5) of the ADE. The measurements ν_i taken by the UAVs' onboard sensors at a location $\mathbf{p}_i = (x_i, y_i, z_i, t_i)$ are modeled as

$$\nu_i = C_i(\boldsymbol{\theta}_{\text{true}}) + \epsilon_i, \qquad i = 1, 2, \ldots, m \,, \tag{3.23}$$

where $C_i(\boldsymbol{\theta}_{\text{true}}) = C(\boldsymbol{\theta}_{\text{true}}, \mathbf{p}_i)$ as in (3.5) and $\boldsymbol{\theta}_{\text{true}}$ is the true parameter vector to be estimated. The measurement error ϵ_i is assumed to be white Gaussian noise with mean $\mathrm{E}(\epsilon_i) = 0$ and variance $\mathrm{var}(\epsilon_i) = \sigma_i^2$. Following the problem definition in [34], the standard deviation σ_i of the error is modeled as $\sigma_i = \frac{\nu_i}{\alpha}$ with a constant signal to noise ratio (SNR) $\alpha^2 = 1000$. It is further assumed that the measurement errors are uncorrelated, i.e. $\mathrm{E}(\epsilon_i \epsilon_j) = \mathrm{cov}(\epsilon_i, \epsilon_j) = 0$ for $i \neq j$.

The following *weighted nonlinear least squares* problem is solved in order to estimate the parameter vector $\boldsymbol{\theta}$:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \left\| \mathbf{W}(\boldsymbol{\nu} - \mathbf{C}(\boldsymbol{\theta})) \right\|_2^2 . \tag{3.24}$$

Here, $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_m)^T$ and $\mathbf{C}(\boldsymbol{\theta}) = (C_1(\boldsymbol{\theta}), C_2(\boldsymbol{\theta}), \ldots, C_m(\boldsymbol{\theta}))^T$. Since the standard deviations σ_i of the measurement errors are not identical, each residual $\nu_i - C_i(\boldsymbol{\theta})$ needs to be weighted by the reciprocal of σ_i. Hence, $\mathbf{W} = \mathrm{diag}(\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_m^{-1}) \in \mathbb{R}^{m \times m}$.

3.4.2 Information Matrix

As detailed in Section 3.3.3, optimum designs for the nonlinear Gaussian puff model (3.5) will always depend on the parameters $\boldsymbol{\theta}$ to be estimated. For a good design, a good estimate $\bar{\boldsymbol{\theta}}$ is required, and vice versa. In order to deal with this interdependence, a sequential design scheme will be applied that is based on a linearization of (3.5) about the current parameter estimate. By Taylor series expansion of (3.5) about $\bar{\boldsymbol{\theta}}$ one obtains

$$C(\boldsymbol{\theta}, \mathbf{p}) = C(\bar{\boldsymbol{\theta}}, \mathbf{p}) + \nabla C(\bar{\boldsymbol{\theta}}, \mathbf{p})^T (\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}) + \ldots \,, \tag{3.25}$$

where

$$\nabla C(\bar{\boldsymbol{\theta}}, \mathbf{p})^T = \left(\frac{\partial C(\boldsymbol{\theta}, \mathbf{p})}{\partial \theta_1}, \frac{\partial C(\boldsymbol{\theta}, \mathbf{p})}{\partial \theta_2}, \ldots, \frac{\partial C(\boldsymbol{\theta}, \mathbf{p})}{\partial \theta_6} \right) \Bigg|_{\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}} \tag{3.26}$$

is the vector of *parameter sensitivities* for the measurement at a location $\mathbf{p}$.

Summing up all measurement data collected at $\mathbf{p}_1, \ldots, \mathbf{p}_m$ yields

$$\mathbf{C}(\boldsymbol{\theta}) = \mathbf{C}(\bar{\boldsymbol{\theta}}) + \mathbf{J}_C \cdot (\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}) \,. \tag{3.27}$$

The Jacobian of $\mathbf{C}(\bar{\boldsymbol{\theta}})$ denoted as

$$\mathbf{J}_C = (\nabla C_1(\bar{\boldsymbol{\theta}}), \ldots, \nabla C_m(\bar{\boldsymbol{\theta}}))^T = (\nabla C(\bar{\boldsymbol{\theta}}, \mathbf{p}_1), \nabla C(\bar{\boldsymbol{\theta}}, \mathbf{p}_2) \ldots, \nabla C(\bar{\boldsymbol{\theta}}, \mathbf{p}_m))^T \tag{3.28}$$

is the so called *extended design matrix* for the linearized model (3.27).

From that, the information matrix

$$\mathbf{M} = \mathbf{J}_C^T \mathbf{W}^2 \mathbf{J}_C \ , \tag{3.29}$$

is obtained with $\mathbf{W}$ as in (3.24). It represents the information content in the measurements $(\mathbf{p}_1, v_1), (\mathbf{p}_2, v_2), \ldots, (\mathbf{p}_m, v_m)$ gathered up to the present point in time.

3.4.3 Optimum Design Problem

The sequence $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m$ of previous measurement locations can be interpreted as an m-trial exact design for the linearized model (3.27)

$$\xi_m = \begin{Bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_m \\ 1/m & 1/m & \ldots & 1/m \end{Bmatrix}$$

and $\mathbf{M}$ from (3.29) relates to the information matrix of ξ_m as $\mathbf{M} = m \cdot \mathbf{M}(\xi_m)$.

We now seek to augment ξ_m to a design with $m + n_V \cdot n_w$ trials by computing sequences of n_w new measurement locations $\tilde{\mathbf{p}}_{v,1}, \ldots, \tilde{\mathbf{p}}_{v,n_w}$ as waypoints for each mobile sensor platform (e.g. UAV) $v = 1, 2, \ldots, n_V$. For this purpose, an information matrix is required that not only accounts for the utility of previous, but also of potential future measurements. Therefore $\mathbf{M}$ is extended in accordance with (3.22), such that

$$\mathbf{M}_{\text{ext}} = \mathbf{M} + \nabla C(\bar{\boldsymbol{\theta}}, \tilde{\mathbf{p}}_{v,1}) \nabla C(\bar{\boldsymbol{\theta}}, \tilde{\mathbf{p}}_{v,1})^T + \cdots + \nabla C(\bar{\boldsymbol{\theta}}, \tilde{\mathbf{p}}_{v,n_w}) \nabla C(\bar{\boldsymbol{\theta}}, \tilde{\mathbf{p}}_{v,n_w})^T \tag{3.30}$$

for $v = 1, 2, \ldots, n_V$. That way, $\mathbf{M}_{\text{ext}}$ becomes dependent on the waypoints to be determined. Minimization of one of the optimality criteria Ψ introduced in Section 3.3.3 subject to the UAVs' motion dynamics models leads to the following design problem for the determination of optimized vehicle-specific waypoint sequences:

$$\begin{aligned}
\min_{\tilde{\mathbf{p}}_{v,1}, \tilde{\mathbf{p}}_{v,2}, \cdots \tilde{\mathbf{p}}_{v,n_w}} \quad & \Psi(\mathbf{M}_{\text{ext}} + \eta \mathbf{I}) && \text{(3.31a)} \\
\text{s.t.} \quad & \mathbf{x}^{k+1} = \mathbf{x}^k + \Delta t_m \cdot \mathbf{f}(\mathbf{x}^k, \mathbf{u}^k) && \text{(3.31b)} \\
& \tilde{\mathbf{p}}_{v,k+1} = \mathbf{C}_v \mathbf{x}^{k+1} && \text{(3.31c)} \\
& \mathbf{x}_{\min} \leq \mathbf{x}^k \leq \mathbf{x}_{\max} && \text{(3.31d)} \\
& \mathbf{u}_{\min} \leq \mathbf{u}^k \leq \mathbf{u}_{\max} \ , && \text{(3.31e)}
\end{aligned}$$

where $k = 0, 1, \ldots, n_w - 1$. Constraint (3.31b) represents the dynamics of all n_V UAVs considered in the problem:

$$\mathbf{x}^k = \begin{pmatrix} \mathbf{x}_1^k \\ \vdots \\ \mathbf{x}_{n_v}^k \end{pmatrix}, \quad \mathbf{u}^k = \begin{pmatrix} \mathbf{u}_1^k \\ \vdots \\ \mathbf{u}_{n_v}^k \end{pmatrix}, \quad \mathbf{f}(\mathbf{x}^k, \mathbf{u}^k) = \begin{pmatrix} \mathbf{f}_1(\mathbf{x}_1^k, \mathbf{u}_1^k) \\ \vdots \\ \mathbf{f}_{n_V}(\mathbf{x}_{n_V}^k, \mathbf{u}_{n_V}^k) \end{pmatrix},$$

where $\mathbf{x}_v, \mathbf{u}_v, \mathbf{f}_v$ correspond to the type of vehicle $v = 1, 2, \ldots, n_V$ and its discretized motion dynamics model, i.e. (2.24) for quadrotor UAVs and (2.28) for fixed-wing UAVs, respectively. The superscript k relates to the time step $t_k = k \cdot \Delta t_m$. By constraining the design region Ξ in that way, the waypoint sequences are tailored to the UAVs individual motion capabilities starting from the initial state $\mathbf{x}_v^0$.

Constraint (3.31c) is not implemented, but stated here to illustrate the extraction of waypoints from the overall vehicle state vector. The term $\eta\mathbf{I}$ with small η is used to regularize the information matrix in order to avoid singularity due to numerical inaccuracies. The software package SNOPT 7.5 [52] is employed to solve NLP (3.31).

3.4.4 Sequential Design Procedure

The overall sequential adaptive sensing procedure can be summarized by Algorithm 1.

Algorithm 1: Sequential adaptive sensing procedure.

select estimate $\bar{\boldsymbol{\theta}}_0$ and a starting position $\mathbf{p}_v$ for each UAV $v = 1, 2, \ldots, n_v$;
take measurements at starting positions $\mathbf{p}_v$;
$k = 0$;
$\tau_0 = t_0$;
$\mathbf{M}_0 = 0$;
while $\tau_k < t_{end}$ **do**
  linearize $\mathbf{C}(\boldsymbol{\theta})$ about $\bar{\boldsymbol{\theta}}_k$: $\mathbf{C}(\boldsymbol{\theta}) = \mathbf{C}(\bar{\boldsymbol{\theta}}_k) + \mathbf{J}_C^k \cdot (\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}_k)$;
  compute information matrix for previous measurements: $\mathbf{M}_k = \mathbf{J}_C^{kT} \mathbf{W}^2 \mathbf{J}_C^k$;
  extend $\mathbf{M}_k$ to desired number of waypoints n_w for each vehicle v:
  $\mathbf{M}_{k+1} = \mathbf{M}_k + \nabla C(\bar{\boldsymbol{\theta}}_k, \tilde{\mathbf{p}}_{v,1}) \nabla C(\bar{\boldsymbol{\theta}}_k, \tilde{\mathbf{p}}_{v,1})^T + \cdots + \nabla C(\bar{\boldsymbol{\theta}}_k, \tilde{\mathbf{p}}_{v,n_w}) \nabla C(\bar{\boldsymbol{\theta}}_k, \tilde{\mathbf{p}}_{v,n_w})^T$;
  determine new waypoints $\tilde{\mathbf{p}}_{v,1}, \ldots, \tilde{\mathbf{p}}_{v,n_w}$ as solution of (3.31) starting from $\mathbf{x}^0 = \mathbf{x}(\tau_k)$;
  let each UAV v collect measurements at $\tilde{\mathbf{p}}_{v,1}, \ldots, \tilde{\mathbf{p}}_{v,n_w}$;
  if *overall number of measurements* $\geq$ *number of parameters in* $\boldsymbol{\theta}$ **then**
    compute new parameter estimate $\bar{\boldsymbol{\theta}}_{k+1}$ as solution of (3.24);
  end
  $\tau_{k+1} = \tau_k + n_w \cdot \Delta t_m$;
  $k = k + 1$;
end

3.4.5 Decentralization

Our high-level objective is the development of a fully decentralized dynamic data-driven control loop (as depicted in Figure 3.4) for cooperative process estimation by multiple UAVs. Therefore, all loop components including PARAMETER ESTIMATION and WAYPOINT CALCULATION have to be performed by each UAV individually. For this purpose, it is assumed that the UAVs exchange the

measurement data they gathered, their current parameter estimate $\bar{\boldsymbol{\theta}}$ as well as their current state vector whenever they are within communication range to each other.

Two options can be considered for the way a UAV calculates its waypoint sequences.

Option 1

The problem (3.31) is set up only for the UAV itself, i.e. constraint (3.31b) contains only one motion dynamics model and only one waypoint sequence is determined. That way, each UAV will try to improve the parameter estimate without considering its teammates' actions.

Option 2

The problem (3.31) is set up for the UAV itself plus all teammates within communication range. From the resulting waypoint sequences, only the calculating UAV's own sequence is actually used. This joint waypoint calculation can be assumed to result in better cooperative behavior than Option 1 since the optimization accounts also for the teammates' (potential) future measurements.

It is obvious that the NLP for the joint waypoint calculation in Option 2 is more complex and its solution, therefore, more time consuming than Option 1. The differences in the UAVs' behavior and the quality of the parameter estimate resulting from the two different options will be evaluated in Section 3.5.3.

3.5 Simulation-Based Validation

The contents of this section have been published in [49].

The purpose of the following analysis is to prove the general effectiveness of the waypoint approach with respect to the quality of the resulting parameter estimate. This is done under idealized conditions omitting inaccuracies stemming from the deviation between the motion dynamics models in problem (3.31) and the actual UAV motion. That means, the vehicle control step in the dynamic data-driven sensing loop (Figure 3.4) is skipped and the sensors are assumed to collect measurements precisely at the calculated spatiotemporal locations. Moreover, perfect communication is assumed, making identical sensor data and parameter estimates available to all UAVs at any time. Measurement noise is the only considered error source.

3.5.1 Comparison of Optimality Criteria

In Section 3.3.3, the criteria for A-, D-, and E-optimality were introduced. From a theoretical point of view, D-optimality has a significant advantage over the other two criteria as it is invariant to affine transformations of $\nabla C(\boldsymbol{\theta}, \mathbf{p})$. Hence, changing a unit of measure, e.g. from [m] to [cm], would not affect the D-optimum design. In order to evaluate differences in the performance of the proposed waypoint approach with respect to the employed optimality criterion, a set of 60 simulation runs per criterion was performed.

parameter	value	unit
domain size	$F = 500$	[m]
min UAV altitude	$z_{\min} = 5$	[m]
initial UAV positions	$\mathbf{x}_1^0 = (-50, 0, 5)$ $\mathbf{x}_2^0 = (-50, -50, 5)$ $\mathbf{x}_3^0 = (-50, 50, 5)$	[m]
sensing rate	$\Delta t_m = 2$	[s]
true puff parameters	$\boldsymbol{\theta}_{\text{true}} = (Q, K_x, x_0, y_0, z_0, t_0)$ $= (1000, 12, 2, 5, 0, 0)$	([kg], [m^2/s], [m], [m], [m], [s])
initial estimate	$\bar{\boldsymbol{\theta}}_0 = (700, 20, 40, 25, 1, 30)$	([kg], [m^2/s], [m], [m], [m], [s])
diffusivity parameter	$K_z = 0.2113$	[m^2/s]
quadrotor UAV		
max x/y/z velocity	$v_{\max} = 10$	[m/s]
max x/y/z acceleration	$u_{\max} = 3$	[m/s^2]
fixed-wing UAV		
max z velocity	$v_{z\max} = 2$	[m/s]
max ang. velocity	$\omega_{\varphi\max} = 9$	[deg/s]
max z acceleration	$u_{z\max} = 0.16$	[m/s^2]
max ang. acceleration	$u_{\varphi\max} = 0.3$	[deg/s^2]
forward speed	$s = 18$	[m/s]

Table 3.1: Simulation parameters used during validation of the developed waypoint approach.

All values of the variables mentioned in the following are given in Table 3.1.

On a $[-F, F] \times [-F, F] \times [0, 50] \subset \mathbb{R}^3$ domain, a Gaussian puff with parameters $\boldsymbol{\theta}_{\text{true}}$ is to be identified. With the joint waypoint calculation approach as introduced in Section 3.4.5 and each of the criteria for A-, D-, and E-optimality, sequences of $n_w = 4$ waypoints per optimization are generated for $n_V = 1/2/3$ quadrotor UAVs starting at the initial positions $\mathbf{x}_{1/2/3}^0$. The last waypoint of the previous sequence serves as starting point for the next waypoint sequence. In order to guarantee that a feasible follow-up sequence exists, $n_w + 2 = 6$ waypoints are determined per optimization, but only the first 4 are actually used.
The UAVs start at time $t = 100$ and take measurements every Δt_m s. Measurement noise is modeled as described in Section 3.4.1 and the wind conditions were uniformly varied between $v_{wx} \in \{0.5, 0, -0.5, 1\}$. The initial parameter estimate $\bar{\boldsymbol{\theta}}_0$ is assumed to be updated every Δt_m s and the sequence of estimates over a period of 50 s was evaluated.

Figure 3.5 shows the resulting RMSE for the different criteria and numbers of UAVs n_V. It reveals, that in terms of the RSME, E-optimality performs significantly worse than A- and D-optimality. They provide very similar estimate accuracies for all numbers of UAVs, D-optimality being slightly superior to A-optimality. Moreover, the different criteria perform differently in terms of the accuracies of the single parameters in $\boldsymbol{\theta}$ as shown in Figure 3.6. Hence, the choice

of the criterion can be made situation-dependent based on the importance of certain parameters. For the following experiments, we will employ the D-optimality criterion as it promises the best overall performance and, in addition, has the useful scaling property.

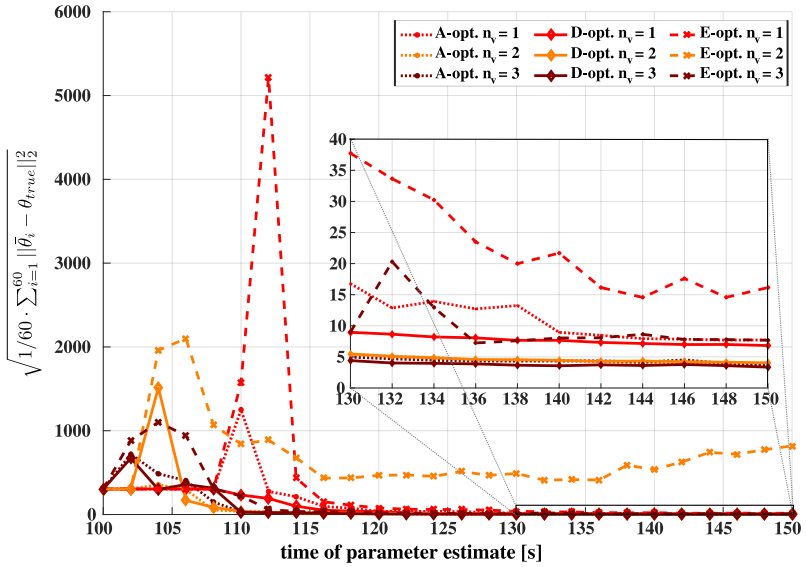


Figure 3.5: RMSE in 60 simulation runs of the joint waypoint calculation approach with $n_v = 1, 2, 3$ quadrotors employing the A-, D-, and E-optimality criterion, respectively. Wind conditions uniformly varied between $u \in \{0.5, 0, -0.5, 1\}$ [m/s].

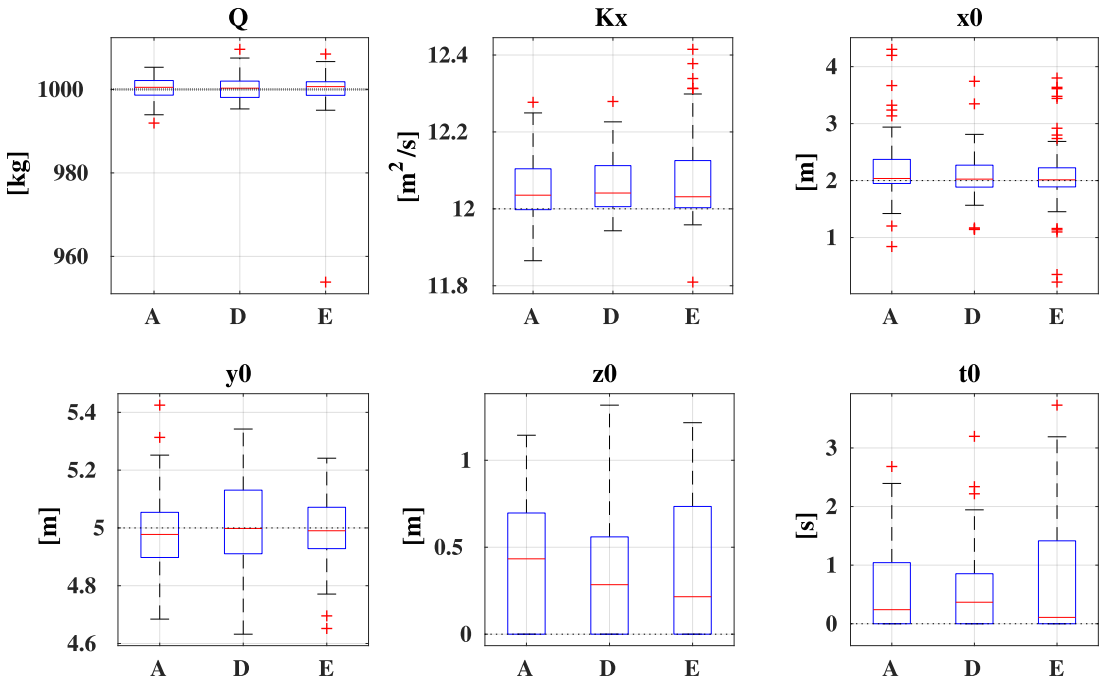


Figure 3.6: Boxplots of the estimated values of the single parameters in $\boldsymbol{\theta}$ at $t = 150$ s in 60 simulations of the joint waypoint approach for $n_V = 3$ employing A-, D-, and E-optimality.

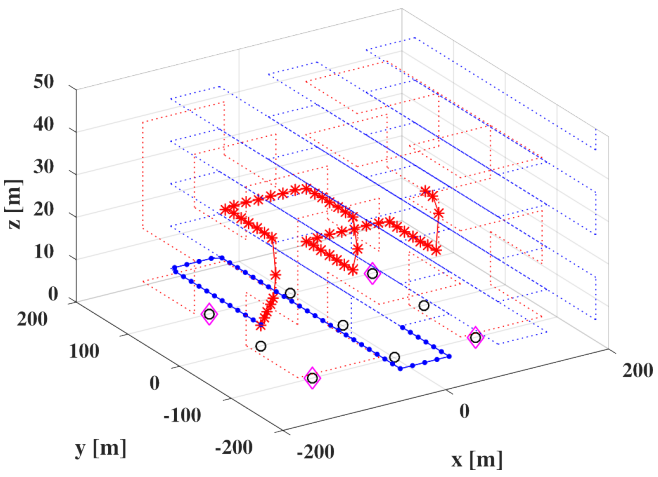


Figure 3.7: Sequences of measurement locations resulting from wavy line motion (blue •) and space-filling Hilbert curve motion (red ∗) over a period of 50 s and $\Delta t_m = 2$ s. The stationary 4-sensor network is depicted as magenta ♢, the 9-sensor network as black ∘.

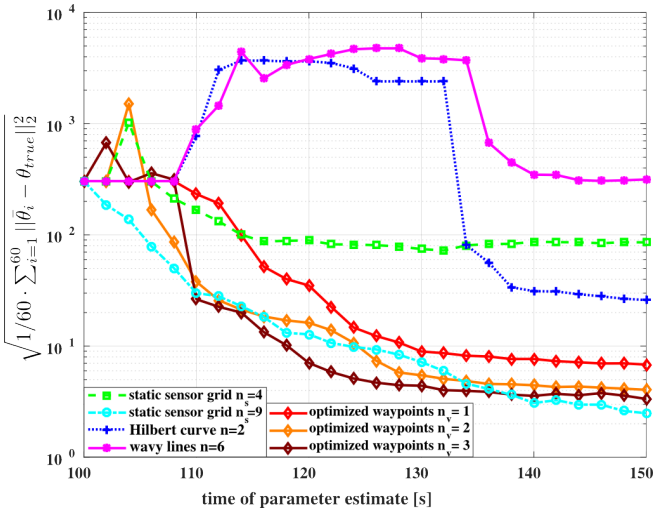


Figure 3.8: RMSE in 60 simulation runs for the waypoints approach with 1/2/3 quadrotors in comparison to a single sensor moving along wavy lines and Hilbert curves as well as to two stationary sensor networks. Wind velocity and direction, respectively, were uniformly varied between $v_{wx} \in \{0.5, 0, -0.5, 1\}$ [m/s].

3.5.2 Comparison to Motion Patterns and Fixed Sensors

In order to prove the effectiveness of following the optimized waypoints for maximizing the informativeness of measurements, the proposed approach is compared to two predefined sensor motion patterns as well as to two stationary sensor networks. In addition to the simulation setup described in Section 3.5.1, a single UAV following a simple wavy line motion pattern and a space-filling Hilbert curve (order $n = 2$), respectively, each starting from $\mathbf{x}_1^0$ is considered. Both patterns are illustrated in Figure 3.7. In all cases, the UAVs start at time $t = 100$ and take measurements every Δt_m s. Additionally, measurements provided by stationary sensor networks with 4 and 9 sensors, respectively, at rate Δt_m are considered. The locations of these sensors are also marked in Figure 3.7. For each sensing approach, the same initial parameter estimate $\bar{\boldsymbol{\theta}}_0$ is considered and updated every Δt_m s. The resulting RMSE curves are shown in Figure 3.8. As can be seen, with respect to estimation error reduction, measurements taken at the optimized waypoints are significantly more effective than measurements taken along predefined sensor paths. Since new waypoints are calculated every 8 s based on the current parameter estimate, the motion of the waypoint-guided UAVs adapts to the current state of the Gaussian puff. Figure 3.9 shows examples of typical waypoint sequences.

While the adaptive sensor motion quickly reduces the RMSE to values < 50, the wavy line motion seems to be especially disadvantageous. Since the source of the puff release is located at $(2, 5, 0)$, lots of measurements taken along the wavy line are located outside the puff. Hence, their information content is low, which has a strong negative influence on the parameter estimate.

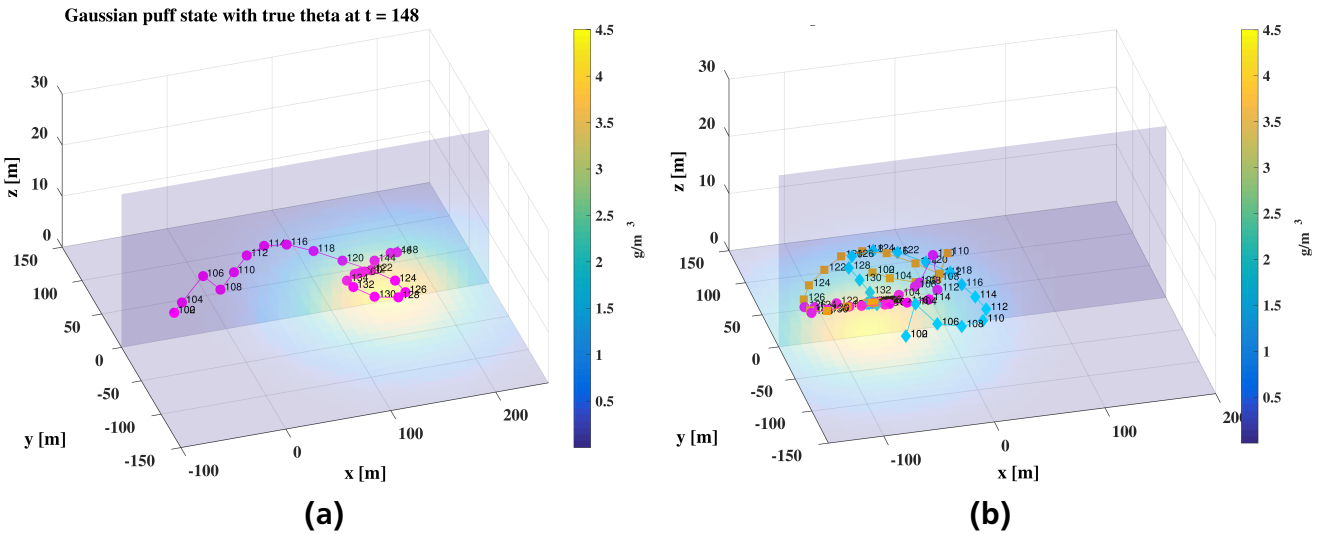


Figure 3.9: Examples of optimized trajectories computed for 1 quadrotor at wind speed $v_{wx} = 1$ **(a)** and 3 quadrotors at $v_{wx} = -0.5$ **(b)**.

For the first estimate update, corresponding to the number of unknown parameters, 6 measurements are required. This is why the curves of the wavy line, Hilbert, and waypoint-guided motion for a single UAV remain constant until enough measurements have been collected. Two and three waypoint-guided UAVs and the stationary sensor networks deliver 2/3/4/9 measurements per time step and therefore the corresponding RMSE descends earlier. As expected, the efficiency of the waypoint approach increases with the number of UAVs. In any case, the final estimation accuracy of the waypoint approach outperforms the 4-sensor network, proving that fewer measurements at optimized locations can be more valuable than more measurements taken at fixed positions. Also the 9-sensor network is temporarily outperformed by the waypoint approach with 3 UAVs, but eventually, the network's RMSE further decreases. This is due to the advantageous positions of the stationary sensors with respect to the puff's evolution towards the end of the simulated time period. The 9-sensor network is then able to gather 9 meaningful measurements per time step in contrast to 3 obtained from the UAVs. This could be completely different in settings where the puff cannot be covered by the fixed sensors. Then flexibly adapting UAVs are clearly in favor.

3.5.3 Comparing Individual and Joint Waypoint Calculation

In order to compare the two waypoint calculation options introduced in Section 3.4.5, 50 simulation runs for each variant were performed for 2 quadrotor UAVs. The wind speed was fixed at $v_{wx} = 0.5$, otherwise, the same simulation setup as before was used (see Table 3.1). Figures 3.10 and 3.11 show typical examples of waypoint sequences obtained from the joint calculation (Option 2) and the individual calculation (Option 1), respectively. Option 1 leads to an alignment of both UAV trajectories while Option 2 distributes the UAVs in the domain. Since nearly all waypoints are at height $z = 5$, only a top-down view is given. As expected, the joint calculation provides a better reduction of the RMSE of the parameter estimate, the only exception being the first waypoint sequence, see Figure 3.12. The average computing time (Dual Core CPU,

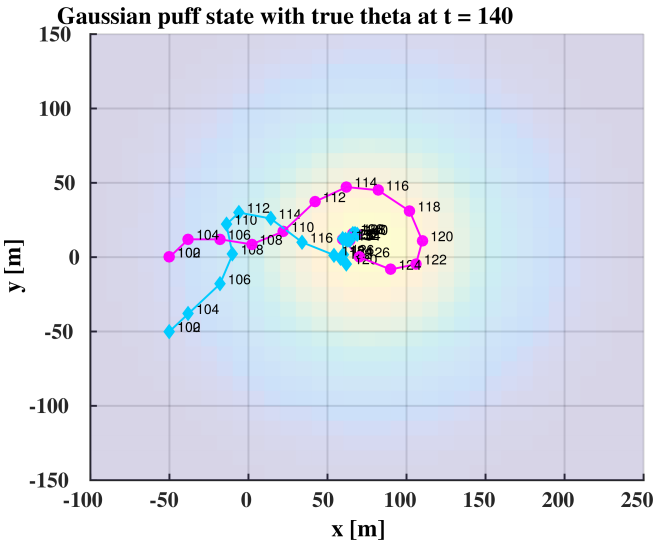


Figure 3.10: Example result of a joint waypoint calculation for 2 quadrotors.

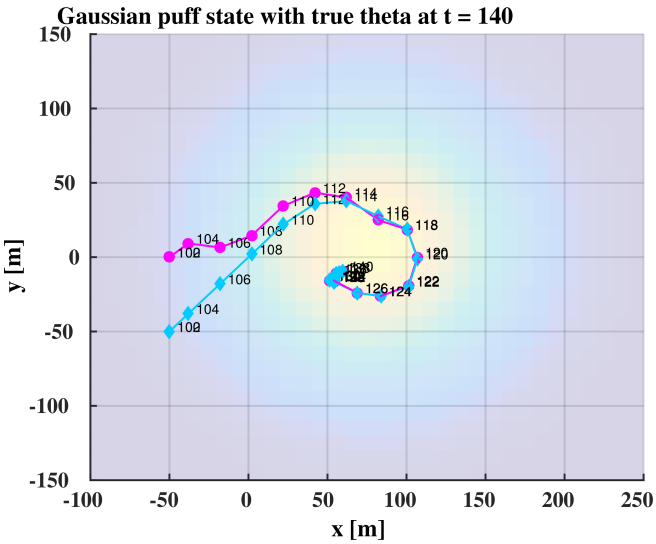


Figure 3.11: Example result of an individual waypoint calculation for 2 quadrotors.

2.53 GHz, 8GB RAM) for one waypoint optimization accounting for both UAVs was 1.098 s, for the individual calculation it was 0.432 s. In a setup with 3 quadrotors, the joint calculation took 4.519 s in average. It is left to the user to decide whether a slightly better performance of Option 2 is worth the significantly higher computational effort. However, Option 2 is preferable if additional constraints, such as collision avoidance, affecting the UAVs' cooperation are to be considered in the waypoint calculation. In order to still be able to scale the approach to teams with $n_V \gg 3$ UAVs without increasing their computational effort, an upper bound for the number of UAVs considered in problem (3.31) along with rules for their selection can be defined.

Another possibility for influencing frequency and effort of the waypoint computation is by varying the number of waypoints n_w per optimized sequence. Figure 3.13 shows the RMSE performance for the simulation setup of Section 3.5.1 with 1 quadrotor and $n_w = 2/4/8$. While $n_w = 8$, i.e. a recomputation interval of 16 s, does not allow good adaptation of the parameter estimate to the gathered data, a recomputation every $n_w = 2$ waypoints seems to be too frequent since not enough relevant additional measurements can be incorporated to improve the parameter estimate. Therefore, $n_w = 4$ appears to be a good tradeoff in terms of estimation error reduction. In terms of computational effort, the average waypoint calculation time for $n_w = 2$ was 0.233 s, for $n_w = 4$ it was 0.675 s, and 1.166 s for $n_w = 8$.

3.5.4 Heterogeneous Teams of UAVs

The proposed waypoint approach is able to deal with heterogeneous vehicle teams as the problem (3.31) can be modularly assembled according to the current UAV constellation. An example of waypoint sequences for a quadrotor and a fixed-wing UAV (with specifications as given in Table 3.1) is shown in Figure 3.14. Due to the fixed-wing UAV's constant speed and its large turning radius, it is less agile and flexible than the quadrotor but can quickly cover signifi-

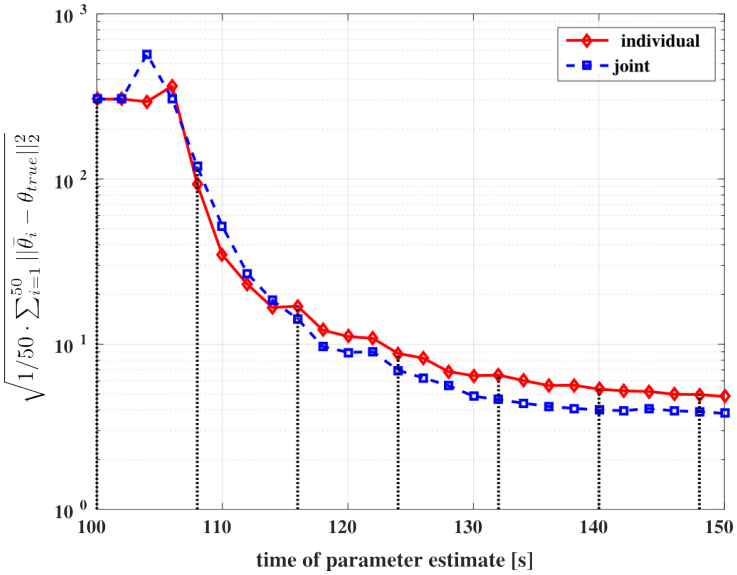


Figure 3.12: RMSE resulting from individual and joint waypoint calculation. Dotted black lines indicate when waypoints were calculated.

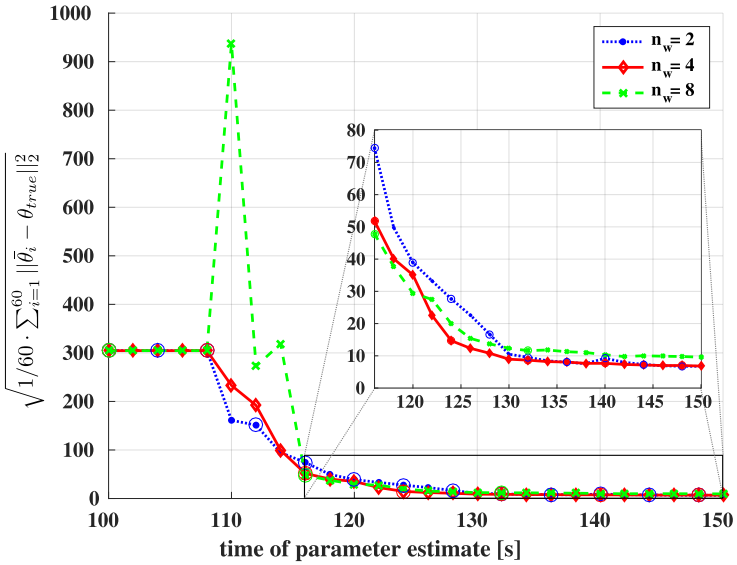


Figure 3.13: RMSE in 60 simulation runs of the waypoint calculation for 1 quadrotor with $n_w = 2, 4, 8$ waypoints per sequence. Circles indicate when new waypoints were computed.

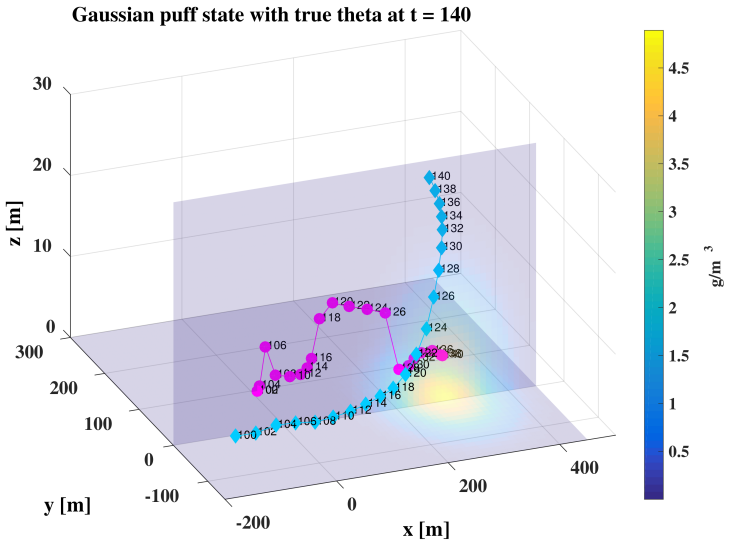


Figure 3.14: Example of waypoints for a quadrotor (magenta) and a fixed-wing (cyan) UAV collecting measurements at wind speed $v_{wx} = 2$ [m/s].

cantly larger domains. This has to be considered when both UAV types are to jointly identify a dispersion process.

A heterogeneous team constellation might be beneficial in cases where a large domain of interest has to be traversed (by a fixed-wing UAV), but at the same time, certain areas require a high local measurement resolution. The latter can best be provided by a quadrotor UAV or a ground vehicle, as these vehicle types can more easily vary their speed or even stop at a certain location.

The proposed adaptive sensing scheme offers the possibility to exploit a team's cooperative sensing activity, independent of the type or number of employed vehicles.

3.6 Summary

An efficient decentralized data-driven sensing strategy for cooperating sensor-equipped UAVs was presented in this chapter. The core is a novel sequential optimum design-based approach for maximizing the informativeness of measurements by computing waypoints that are individually tailored to each vehicle and exploit the team's cooperative mobility (Section 3.4). Measurements at the optimized waypoint locations lead to the best possible improvement of the parameter estimate for a Gaussian puff model of an atmospheric dispersion process. Simulation results illustrate the proposed approach from different perspectives (Section 3.5), e.g. regarding the choice of a suitable OED optimality criterion (Section 3.5.1), different decentralization options or waypoint sequence lengths (Section 3.5.3), and its ability to handle heterogeneous team configurations (Section 3.5.4). The effectiveness of the adaptive sensing scheme is demonstrated in comparison to non-adaptive sensor motion patterns and stationary sensor networks (Section 3.5.2).

The results in Section 3.5 are intended as a general proof of concept under idealized conditions, e.g. ignoring the effects of deviations between the modeled and the actual UAV motion. Hence, the contents of this chapter focus on the optimal parameter estimation aspect of the multi-

vehicle monitoring problem motivated in Chapter 1 and provide the second required component of the overall dynamic data-driven cooperative sensing and control approach proposed in this thesis.

As we now have effective solutions for both problem aspects – cooperative control (Chapter2) and process estimation (Chapter3) – at hand, the consequential next step is their combination and evaluation in a more realistic simulation setup, which is subject of the next chapter.

4 Cooperative Control for Dynamic Data-Driven Multi-Objective Monitoring Tasks

This chapter describes how the solutions for cooperative control and adaptive sensing, respectively, proposed in the previous two chapters are brought together to form a novel dynamic data-driven sensing and control scheme. After a short review of related work (Section 4.1), the contribution to the field is summarized in Section 4.2. Section 4.3 gives further details on the proposed scheme.

The new approach has been evaluated by means of a realistic simulation. For this purpose, a ROS/Gazebo framework has been developed, offering the additional advantage of simplifying a future transfer from simulated to real applications. Its overall setup, as well as some specific components, are briefly described in Section 4.4. Evaluation results for a representative multi-objective monitoring scenario are presented in Section 4.5 followed by two other examples of problems that can be solved by the proposed scheme in Section 4.6.

4.1 Related Work in Cooperative Monitoring of Spatio-Temporal Processes

As could be seen from the literature reviews in Chapters 2 and 3, a lot of effort has been made for developing strategies for cooperative control and adaptive sensing by multiple autonomous vehicles. However, the number of existing publications that deal with multi-objective monitoring tasks combining dynamic data-driven sensing with cooperative mobility of multiple, possibly heterogeneous, sensor vehicles is very limited.

Two examples of integrated monitoring setups involving real unmanned systems hardware, an operator ground station, and communication between the different entities are the AirShield project [37, 38] and the COMETS project [100].
The aim of the AirShield project was data acquisition by a swarm of micro UAVs in the context of atmospheric dispersion of hazardous material. A decentralized bio-inspired swarm behavior is employed to guide the UAVs along a given path while maximizing spatial coverage, but avoiding self-separation and loss of connectivity. While the need for sensor data feedback and adaptive calculation of flight paths was identified in [38], it was not realized within the scope of the project. Instead, the focus was on reliable UAV-to-ground-station and inter-UAV connectivity.
The COMETS project dealt with cooperative forest fire detection and monitoring by multiple heterogeneous UAVs with human operator interaction. Two types of helicopters and an airship were employed at different autonomy levels ranging from full autonomy to teleoperation.

A decentralized heuristic approach combining cooperative control and adaptive sensing is described in [98]. Multiple vehicles are to climb the gradient of an unknown field by moving in

a formation that is adaptively reconfigured in response to the collected data and the changing environment.

To the field of model-aided path planning and sampling for algal bloom tracking, different approaches and field experiments were contributed by the University of Southern California Center for Integrated Networked Aquatic PlatformS (USC CINAPS) [152, 137, 136].
In [152], an algorithm is presented that plans optimal trajectories for a robotic boat based on feedback from a static sensor network and an underlying scalar field model of the bloom that is to be estimated.
Heterogeneous aquatic platforms, namely two underwater gliders, one autonomous surface vehicle, and a number of static buoys, are employed for reconstructing the scalar field in [136]. A central instance adaptively provides waypoints for the vehicles by means of a model-based prediction of the movement of a feature of interest, which is continuously updated by assimilating the gathered measurement data. While heading for the sampling locations, the vehicles are to avoid collisions as well as obstacles and maintain connectivity. This is achieved by null-space-based behavioral control.
It has to be noted that algal blooms evolve at a completely different time scale compared to atmospheric dispersion processes, allowing much longer computation times.

Dynamic Data-Driven Application Systems

The approaches presented in [98], [152], and [136] as well as a number of approaches that were already mentioned at the end of Section 3.1 on adaptive sensing methods [145, 146, 62, 108, 109, 126, 47, 127] can be counted among the so-called Dynamic Data-Driven Application Systems (DDDAS).

The term DDDAS stands for a paradigm first introduced by Frederica Darema [39] and describes application systems that have 1) the ability to incorporate additional data (e.g. collected by sensors) at runtime and 2) the ability to steer the measurement process, i.e. to influence the type or quality of incoming new data. Thus, DDDAS are characterized by a permanent bidirectional interaction, in other words, a feedback control loop, between application (e.g. a simulation) and data collection.

Darema nicely summarizes the advantages of this paradigm as *"Application simulations that can dynamically incorporate new data, archival or from on-line measurements of the actual systems, offer the promise of more accurate analysis, more accurate predictions, more precise controls, and more reliable outcomes."* and *"The ability to guide the measurement process and selectively focus on a subset of the measurement space can result in more efficient and effective measurements, which can be desirable in reducing cost, collection time, or improving on the quality of data collected."*

These properties make dynamic data-driven sensing schemes highly desirable also for the environmental monitoring applications considered in this thesis.

4.2 Contribution

The literature on multi-vehicle monitoring missions combining adaptive sensing and cooperative control tasks is very limited, not to mention literature on DDDAS additionally accounting for

vehicle cooperation. Especially, the coupling of adaptive sensing for parameter estimation and cooperative mobility has hardly been investigated.

Therefore, a novel dynamic data-driven sensing and control scheme is proposed in this chapter that couples the mixed-integer MPC approach as described in Chapter 2 with the sequential optimum design of sensing trajectories as introduced in Chapter 3. It is versatilely applicable in multi-objective monitoring applications for teams of sensor-equipped vehicles that involve cooperative adaptive sensing for optimal parameter estimation of a process model and additional cooperative mobility tasks.

The proposed DDDAS is fully optimization-based, yet efficient enough to be performed online on board each vehicle. Decentrality makes it scalable also to large team sizes. Both the employed waypoint calculation as well as the cooperative MPC explicitly account for and, hence, exploit the vehicles' motion dynamics.

4.3 Proposed Combination of Mixed-Integer MPC and Sequential Optimum Design

Figure 4.1 illustrates the interaction of the different elements the proposed dynamic data-driven control scheme is composed of. All parts are performed decentrally on board each individual vehicle, assuming that team members can share information on their current state, the waypoint they are heading for as well as the measurements they collected whenever they are within communication range to each other. In the following, the elements are briefly described with references to the sections providing more detailed information and their interaction is discussed, starting at the top of the depicted loop and continuing clockwise.

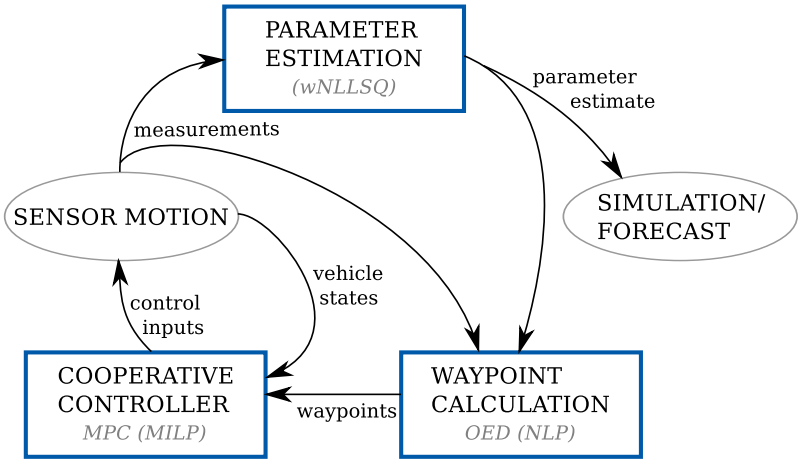


Figure 4.1: Scheme of the overall DDDAS developed in this thesis.

For identifying the parameters of the model used to simulate and forecast the dynamic phenomenon of interest, these parameters are estimated based on the available measurement data by solving the weighted Nonlinear Least Squares (wNLLSQ) problem (3.24) introduced in Section 3.4.1. As described in Section 3.4.2, an information matrix is then derived from a linearization of the model about the current parameter estimate and serves as input for the minimization of a suitable optimality criterion subject to the vehicles' motion dynamics (problem (3.31) described in Section 3.4.3). Thus, a sequence of optimal vehicle-specific waypoints for

maximizing the information gain of future measurements is obtained. The waypoint calculation is decentralized in terms of Option 1 described in Section 3.4.5.

The model-predictive controller (Chapter 2) then provides the corresponding control inputs to optimally guide a vehicle along its waypoints while accounting for additional cooperative mission objectives. The MPC approach by itself forms a sub-loop of the overall repeating feedback procedure. From a control theoretic point of view, the proposed scheme may be interpreted as cascade control system with the cooperative controller acting as inner secondary controller within the outer primary loop for optimal parameter estimation. The inner loop is executed every Δt s. The choice of Δt depends, among practical aspects like the available computing power, on the size and complexity of the underlying MILP model and on the MPC prediction horizon N. A suitable value for N is scenario-dependent. For the applications considered in this thesis, an important factor is if a vehicle has to reach far away target points or operates in a small local environment only. If targets are located far away and N is too small, then a target might not be reachable for a vehicle within the current prediction horizon and it will not move in the desired direction. On the other hand, large values of N might lead to an unnecessary increase in problem complexity and the required solution time. Also, in constantly changing environments, as considered here, longer prediction horizons not necessarily yield better solutions.

As it is assumed that the sensor-equipped vehicles continuously measure every Δt_m s, where $\Delta t_m > \Delta t$, applying the optimized control inputs directly defines where a sensor takes its next measurement. Hence, the inner loop directly influences the outer loop, as required for cascade control. The outer loop is closed by using the new sensor data to update the parameter estimate and, hence, the process simulation. This is done after every measurement, i.e. every Δt_m s. A new waypoint calculation is triggered whenever the timestamps of all except the last element in the previously calculated sequence of n_w waypoints have expired, i.e. the outer loop is executed every $(n_w - 1) \cdot \Delta t_m$ s. That means the inner cooperative control loop is much faster than the outer loop, another requirement of cascade control. The constraints in (3.31) ensure that the computed waypoints are reachable by a UAV within its work domain. Infeasible vehicle states would otherwise cause the current MILP solution in the vehicle's inner MPC loop to fail.

If optimal parameter estimation is the only mission objective, the inner control loop compensates deviations of the vehicles' trajectories from the desired waypoints. Hence, the primary controller remains unaffected by this kind of disturbances that handled by the secondary controller, which is typically an important feature of cascade control type schemes. However, if multiple objectives are pursued by the cooperative controller and the vehicles have other tasks besides following their waypoints, as considered here, it is likely that they, to a certain extent, diverge from their optimized sensing trajectories. That is, the inner cooperative controller "intentionally" disturbs the outer sensing loop. For this reason, each waypoint sequence optimization is initialized with the UAV's current motion state in order to make the most of the actual vehicle behavior. That way, the outer estimation procedure becomes more robust to this kind of disturbance and the proposed DDDAS is able to handle adaptive sensing plus additional cooperative mobility tasks.

4.4 Overview of and Contributions to the Simulation Framework used for Evaluation

Parts of this section were published in [48].

The simulation framework used to evaluate the proposed dynamic data-driven sensing and control loop is implemented employing the Robot Operating System (ROS) ([116], `http://www.ros.org`) and Gazebo ([77], `http://www.gazebosim.org`) as multi-robot simulator. A UAV is represented by a collection of several ROS nodes, each responsible for a certain functionality, that communicate their inputs and outputs via ROS topics or ROS services. The ROS graph in Figure 4.2 shows the ROS components each UAV (in the depicted example denoted as "robot0" and "robot1") is composed of, as well as their interaction via ROS topics. A comprehensive description is omitted here, but a few details worth mentioning are given below.

4.4.1 Selected Implementation Details

The core of each UAV's motion control is the node `/rob` implementing the mixed-integer MPC. During each control cycle, i.e. every Δt s, here, the MILP representing the vehicle's individual subproblem is reassembled at runtime based on the current state of its local environment. This poses a significant advantage over the Matlab implementation used for the applications described in Section 2.5, where the Multi-Parametric Toolbox (MPT) [81] was employed for MPC and controllers for every possible subsystem configuration had to be compiled a priori. Moreover, the MILP is now solved using the faster Gurobi solver [59] instead of CPLEX [68] (cf. Section 2.3.3). Hence, the ROS implementation of the mixed-integer MPC approach is much more flexible and more efficient than its Matlab counterpart.

For efficiency reasons, the quadrotor and fixed-wing UAV motion, respectively, is within the model-predictive controller represented by the basic motion dynamics models introduced in Section 2.4.1. However, the obtained control inputs are, in the case of quadrotors, applied to the comprehensive and realistic Gazebo model `hector_quadrotor_gazebo` [91] for simulating the motion of real quadrotor UAVs. For simulating fixed-wing UAVs, a new Gazebo model has been developed (see Section 4.4.2). The simulation models for both UAV types are, therefore, much more complex than those assumed by the controller and can be seen as proper approximations of real UAVs. This is another improvement compared to the original Matlab framework, where the vehicle simulation employed the same motion dynamics models as the MPC.

Both motion dynamics models as introduced in Section 2.4.1 are of second order. The state vectors contain vehicle positions and velocities while the control vectors contain accelerations. The Gazebo simulation models of both quadrotors and airplanes, though, are controlled via the ROS topic `/cmd_vel`, i.e. they expect velocity commands. Therefore, from the second order trajectory optimization by MPC, the resulting velocity state variables instead of the acceleration controls are returned as command inputs for Gazebo.

Simulations are visualized using the 3D visualization tool Rviz.

The node `/waypointManager`, responsible for managing a vehicle's waypoints for optimized sensing, initializes a new waypoint calculation in virtue of Algorithm 1 (page 58) whenever the UAV is heading for the last waypoint in the previously optimized sequence. In order to keep up

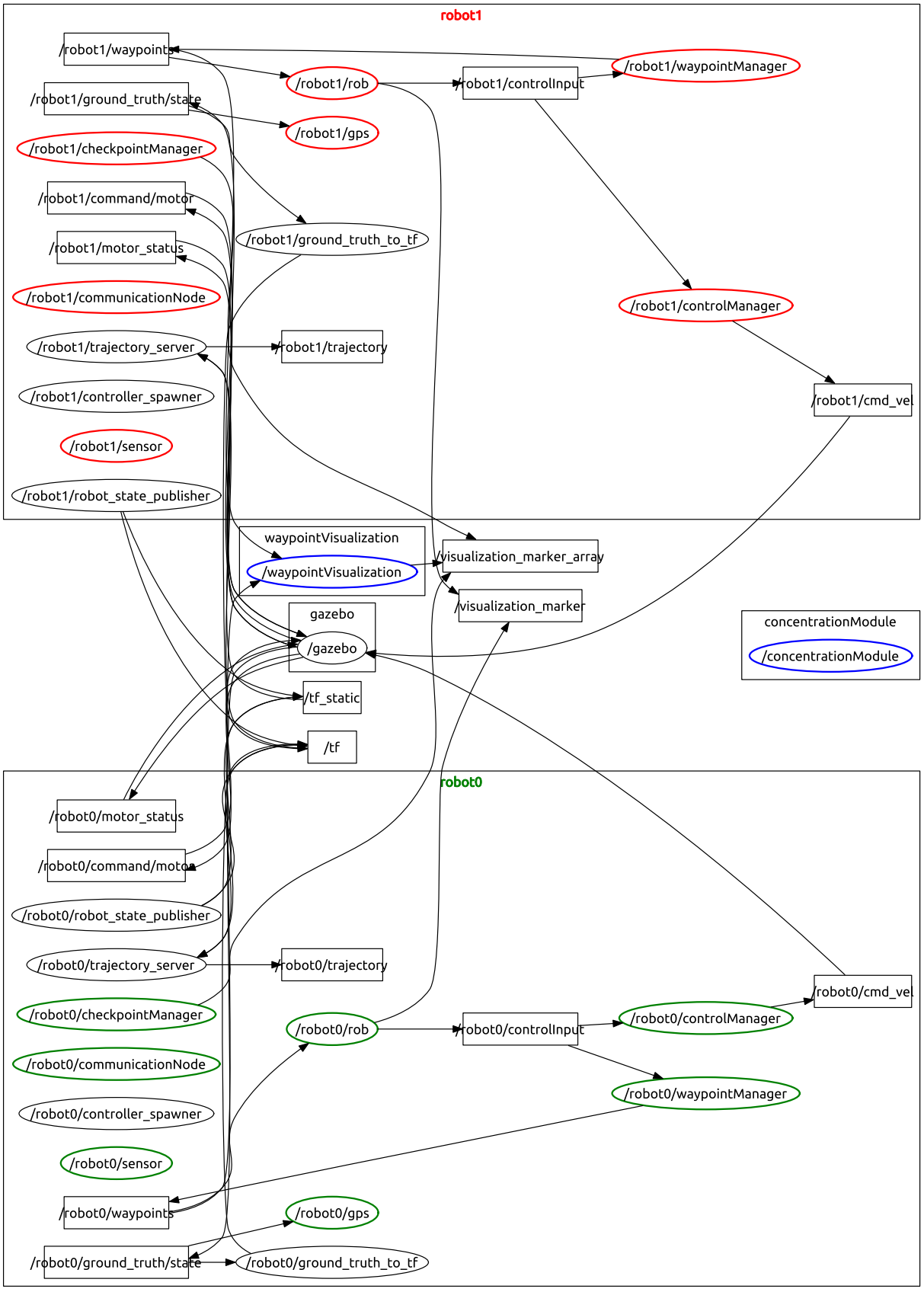


Figure 4.2: ROS graph of the developed simulation framework. Bubbles represent ROS nodes, boxes represent ROS topics. Arrows indicate how nodes publish (outgoing arrow) or subscribe (incoming arrow) to topics. The colored nodes have been implemented within the scope of this thesis. Communication via ROS services is not visualized.

a fluent vehicle motion, based on the current sequence of control inputs provided by the MPC, a prediction of the UAV's state 3 time steps ahead is made and used as initial value for the new waypoint optimization. SNOPT 7.5 [52] is employed to solve problem (3.31).

In order to simulate how a vehicle's onboard sensor takes a measurement, the ROS node `/sensor` calls a ROS service provided by the node `/concentrationModule`. The latter returns a concentration value by evaluating the Gaussian puff model (3.5) and adding a measurement error corresponding to the sensor model described in Section 3.4.1.

Information exchange among the teammates is routed through the node `/communicationNode` that models their communication based on their distances to each other. That means, that ROS service calls from a ROS node belonging to UAV i can only be answered by a node belonging to UAV j's namespace if the two simulated vehicles are within communication range d_{com} to each other. Other connectivity requirements, as well as communication loss or delays, could also be modeled by that node.

4.4.2 Fixed-Wing UAV Simulation in ROS/Gazebo

The contributions summarized in this section were developed in cooperation with Felix Treede. Details can be found in [142].

The physics-based simulator Gazebo is originally not designed for simulation of fluid dynamics, which is, however, essential for simulating the flight of (fixed-wing) aerial vehicles. In order to bridge this gap, Gazebo versions 3.0+ include the LiftDragPlugin [3]. It simulates the lift and drag forces acting on an object that is immersed in a fluid and directly applies them to the object links.

Based on this plugin, a Gazebo simulation model of a fixed-wing UAV was implemented, which comprises

- a 3D airplane model specified in the Unified Robot Description Format (URDF) in order to likewise enable visualization in Gazebo as well as in RViz,
- a Gazebo plugin that receives the desired flight control surface positions and forward thrust and applies them to the simulation model, and
- an autopilot implemented in ROS as an interface between the mixed-integer MPC based on the airplane dynamics model (2.27) and the Gazebo simulation model.

The listed components are briefly described below followed by selected validation results.

Fixed-Wing UAV Model

The developed model is a modified version of an existing demo model for the Gazebo LiftDragPlugin. Its dimensions were scaled down by factor 10 resulting in a small-scale aircraft with a wingspan of 2 m and a weight of 2.6 kg. As can be seen in Figure 4.4, the airplane model is minimalistic, but nevertheless, comprises all components necessary to enable flight. In contrast to real aircrafts, the model does not have separate flight control surfaces, i.e. ailerons, elevators, and rudder as depicted in Figure 4.3 to influence pitch, roll, and yaw, respectively. Instead, the whole wing or surface is rotated in order to save simulation elements.

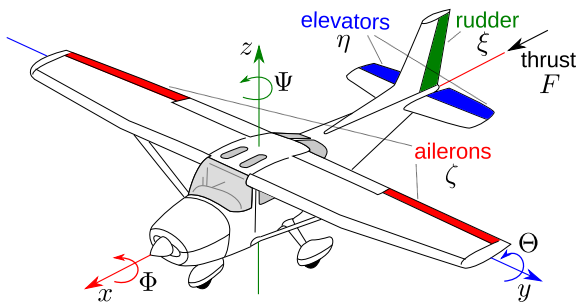


Figure 4.3: Schematic view of an airplane's flight control surfaces and the corresponding control variables of the implemented simulation model.
(Source: modified figure from [4])

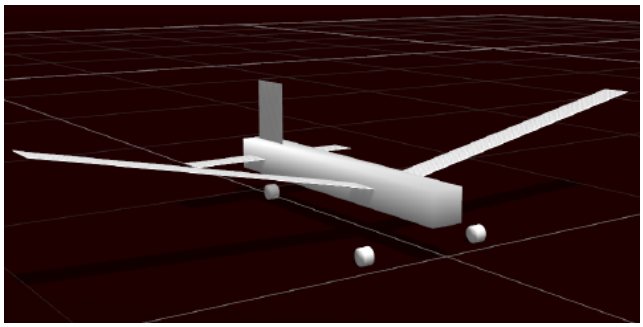


Figure 4.4: Visualization of the developed 3D airplane model in Gazebo.
(Source: [142])

A specifically developed Gazebo plugin receives the desired flight control surface positions as well as forward thrust via ROS topics and applies them to the URDF model.

Autopilot

In order to translate the control inputs (forward speed s, climb/descent rate v_z, and angular velocity ω_φ) as provided by the model-predictive controller based on the airplane model introduced in Section 2.4.1 into flight control surface positions (ζ for ailerons, η for elevators, and ξ for the rudder) and thrust F as expected by the Gazebo model, an interface, called *autopilot*, was implemented in ROS using the concept of ROS nodelets. The general autopilot concept employing cascades of PID controllers is illustrated in Figure 4.5. The variable names correspond to those introduced in Figure 4.3. The controller parameters for the autopilot were determined by extensive experimentation.

Validation

The basic airplane motion dynamics model employed by the mixed-integer MPC differs considerably from the dynamics of the Gazebo simulation model. In order to assess this difference in terms of the UAV's flight path, a reference simulation was set up based on the first order dynamics model

$$\dot{\tilde{\mathbf{x}}}_{fw}(t) = \begin{pmatrix} s \cdot \cos\varphi(t) \\ s \cdot \sin\varphi(t) \\ v_z(t) \\ \omega_\varphi(t) \end{pmatrix} = \tilde{\mathbf{f}}_{fw}(\tilde{\mathbf{x}}_{fw}(t), \tilde{\mathbf{u}}_{fw}(t)) \,, \tag{4.1}$$

where the state vector $\tilde{\mathbf{x}}_{fw}(t) = (x(t), y(t), z(t), \varphi(t))^T$ contains the UAV's x/y/z position and orientation φ and the control vector $\tilde{\mathbf{u}}_{fw}(t) = (s, v_z(t), \omega_\varphi(t))^T$ comprises the constant forward speed s, the climb/descent rate v_z, and the angular velocity ω_φ. Explicit Euler integration was employed. This corresponds to a first order version of (2.28).

For a comparison of the resulting trajectories, identical control inputs s, v_z, and ω_φ were applied to both the reference simulation and the autopilot-controlled Gazebo airplane model. Different

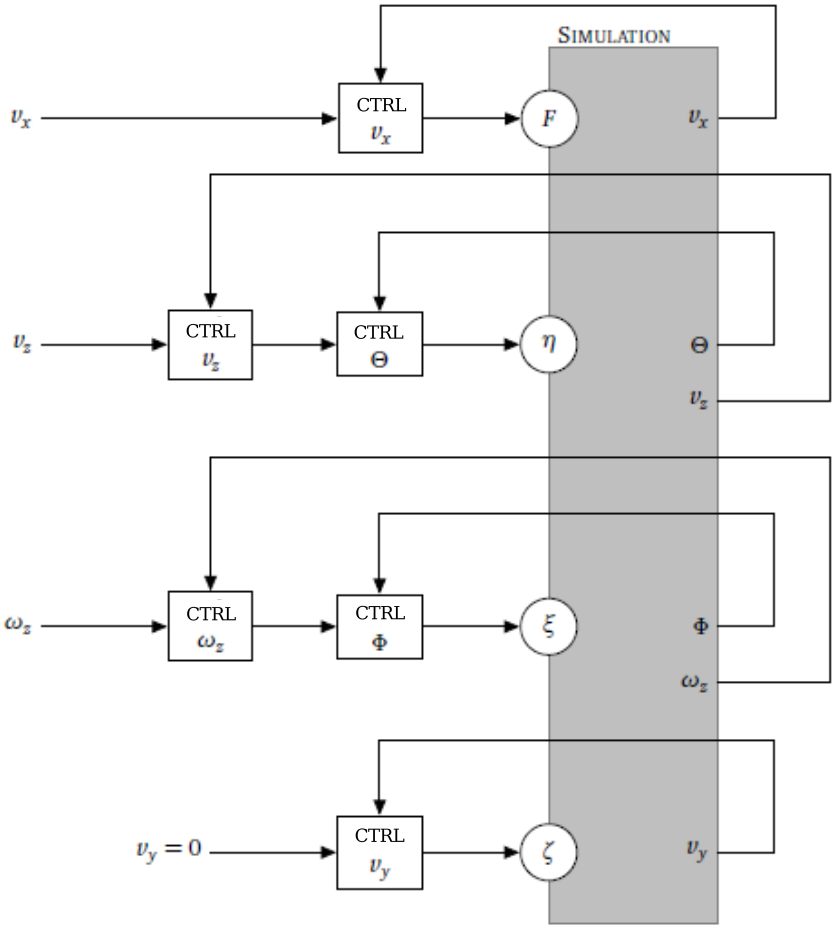


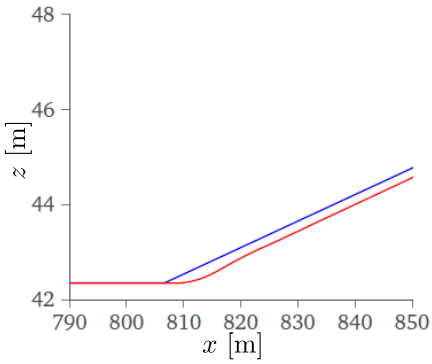
Figure 4.5: Autopilot concept for translating velocity commands provided by the MPC into angle and force inputs expected by the Gazebo airplane model (depicted as grey box). Note that $v_x \triangleq s$ and $\omega_z \triangleq \omega_\varphi$. (Source: [142])

settings have been evaluated, the results are shown in Figure 4.6. It can be concluded that the Gazebo simulation is able to follow the reference path fairly well as long as the velocity inputs remain constant. After a change in velocity, the simulated trajectories deviate from their reference. This is due to the reference simulation immediately applying the new velocity, while the Gazebo simulated airplane has to accelerate first. For this reason, the second order airplane model (2.27) was employed instead of (4.1) for the cooperative monitoring applications described in Section 4.6.

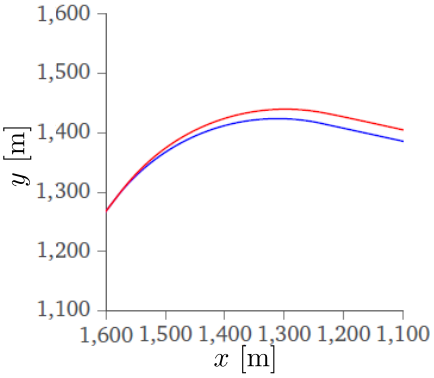
4.5 Application Scenario: Cooperative Parameter Estimation and Patrol

This section summarizes own work published in [48].

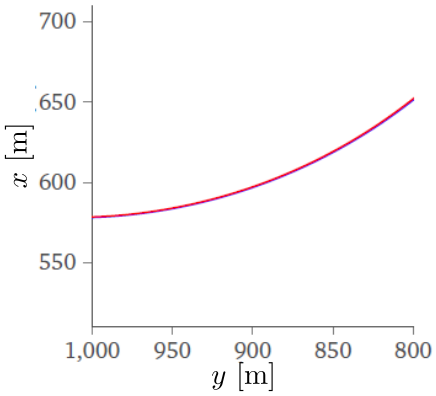
The dynamic data-driven sensing and control scheme as described in Section 4.3 has been applied in a multi-objective monitoring scenario combining adaptive sensing for parameter estimation and cooperative patrol of predefined checkpoint locations. The scenario was simulated using the ROS/Gazebo implementation presented in Section 4.4.



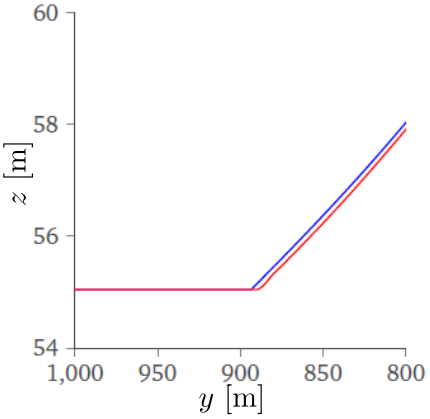
(a) Trajectories after a change of $v_z = 0\frac{\text{m}}{\text{s}}$ to $v_z = 1\frac{\text{m}}{\text{s}}$ during straight flight.



(b) Trajectories after a change of $\omega_\varphi = 0\frac{^\circ}{\text{s}}$ to $\omega_\varphi = 3\frac{^\circ}{\text{s}}$ and reverse.



(c) Trajectories after a change of $v_z = 0\frac{\text{m}}{\text{s}}$ to $v_z = 0.5\frac{\text{m}}{\text{s}}$ while maintaining $\omega_\varphi = 3\frac{^\circ}{\text{s}}$. projected onto the YX plane.



(d) Trajectories after a change of $v_z = 0\frac{\text{m}}{\text{s}}$ to $v_z = 0.5\frac{\text{m}}{\text{s}}$ while maintaining $\omega_\varphi = 3\frac{^\circ}{\text{s}}$. projected onto the YZ plane.

Figure 4.6: Comparison of airplane trajectories based on the reference model (4.1) (blue) and the autopilot-controlled Gazebo simulation (red). In all cases $s = 18\frac{\text{m}}{\text{s}} = const.$ (Source: [142])

4.5.1 Problem Statement

The starting point of the considered application scenario is an instantaneous gas release, the evolution of which is to be predicted based on a Gaussian puff model of the dispersion process. This requires an accurate estimate of the model's parameters, which is obtained and repeatedly improved based on measurement data collected by a team of n_V homogeneous sensor-equipped quadrotor UAVs deployed in a working area $\Omega \in \mathbb{R}^3$. The onboard sensors are assumed to continuously measure every Δt_m s.

The UAVs' one task is to cooperatively patrol n_C a priori defined checkpoints $\in \Omega$ representing locations of special interest, e.g. train stations or hospitals, where accurate local concentration data are required. Checkpoints can be visited in any order at any time. However, in order to increase the incentive for frequent visits, each checkpoint is labeled with a penalty value p_c that increases with each time step t_k according to

$$p_c^{k+1} = p_c^k + \Delta p \ , \quad c = 1, 2, \ldots, n_C \tag{4.2}$$

while the checkpoint remains unvisited. It is reset to zero when a local measurement has been taken. As soon as the gas concentration at a checkpoint falls below a threshold $c_{\max}$, it is marked "done" and removed from the task.

In order to make the UAVs' flight paths most valuable with respect to the improvement of the parameter estimate, their other task is to follow sequences of vehicle-specific waypoints obtained from the sequential optimum design approach (Section 3.4). Figure 4.7 illustrates the scenario combining waypoint following and checkpoint patrol.

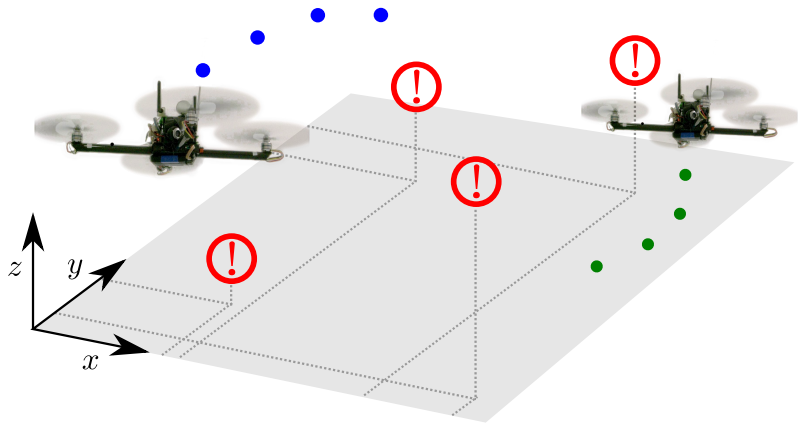


Figure 4.7: Illustration of the considered problem. Sensor-equipped quadrotor UAVs continuously take measurements while following individual optimized waypoints. At the same time, points of special interest are to be visited regularly.

The mixed-integer model-predictive controller (Section 2.4) provides the control inputs to guide the UAVs while trading off the two mission aspects. Controls are updated every $\Delta t = 1\,\text{s}$ and a prediction horizon length $N = 10$ is used. Motion control, waypoint calculation, and checkpoint penalty management is performed by each team member individually. The UAVs are assumed to exchange their current position, their individual measurement data, the checkpoint penalties as well as their current waypoint as long as they are within communication range d_{com} to each other.

4.5.2 Objectives

Each UAV is faced with two conflicting objectives:

1. *Follow the waypoints* for most informative sensing.
2. *Visit the checkpoints* to provide accurate local concentration data. Select checkpoints according to their priority represented as penalties.

4.5.3 Modeling

Quadrotors are modeled as point masses with Euler discretized double integrator dynamics (2.24).

Distances $d^k_{vw_v}$ between a vehicle v and its current waypoint w_v are approximated as described in Section 2.4.2. The same applies to distances d^k_{vc} between UAV v and checkpoint c.

The logical rules expressing whether a checkpoint has been visited by a UAV are modeled analog to (2.35) and (2.36), i.e. binary variables $b^k_{vc} = 1$ indicate if UAV v is currently within distance d_{check} to checkpoint c and binary variables $s^k_c = 0$ indicate if checkpoint c is currently visited by any UAV.

Similar to (2.37), cooperation among the UAVs is realized by minimizing the penalties of those checkpoints currently not visited by any team member. This selection is modeled using the binary variables s^k_c and a set of auxiliary variables $h^k_c \in \mathbb{R}$ that either equal the penalties p^k_c or equal zero, depending on the status of checkpoint c:

$$h^k_c = s^k_c \cdot p^k_c \ . \tag{4.3}$$

Penalty increase over time is modeled as stated in Sec. 4.5.1:

$$p^{k+1}_c = h^k_c + s^k_c \cdot \Delta p \ , \tag{4.4}$$

where p^0_c equals the current checkpoint penalty at the time of the controller call.

In order to enable the formulation of another incentive for checkpoint visits, a variable representing the maximum of all penalties over the prediction horizon N is introduced:t

$$p_m \geq p^k_c \ . \tag{4.5}$$

Logical expressions are transformed into mixed-integer linear constraints employing the Big-M method introduced in Section 2.3.2.

Optimization Criteria

The performance criterion (2.19) of the CFTOC problem forming the basis of the mixed-integer MPC has to represent the trade-off between the two conflicting objectives. The waypoint objec-

tive requires no explicit cooperation as every UAV follows its individual waypoints, which can be represented as minimization of

$$J_1 = \sum_{\nu=1}^{\widetilde{n}_V} \sum_{k=0}^{N} d_{\nu w_\nu}^k \,, \tag{4.6}$$

where $\widetilde{n}_V \leq n_V$ is the number of UAVs in the local subsystem. Modeling the checkpoint objective, however, has to ensure that no two UAVs visit the same checkpoint and has to account for the checkpoints' different priorities. This is achieved by a combination of three criteria:

$$J_2 = \sum_{\nu=1}^{\widetilde{n}_V} \sum_{c=1}^{n_C} \sum_{k=0}^{N} \frac{1}{n_C} \cdot d_{\nu c}^k \,, \tag{4.7a}$$

$$J_3 = \sum_{c=1}^{n_C} \sum_{k=0}^{N} \frac{1}{n_C \cdot p_{\max}^0} \cdot h_c^k \,, \quad \text{and} \tag{4.7b}$$

$$J_4 = \frac{N}{p_{\max}^0} \cdot p_m \,, \tag{4.7c}$$

where $p_{\max}^0 = \max_c p_c^0$. J_2 can be seen as an auxiliary objective preventing the UAVs to drift away from the checkpoints. The factor $\frac{1}{n_C}$ is used to normalize the sum of distances in relation to J_1. J_3 represents the sum of penalties of all unvisited checkpoints over the time horizon N, which is to be minimized. In addition, the maximum penalty p_m denoted as J_4 is to be kept at a minimum. The factors $\frac{1}{n_C \cdot p_{\max}^0}$ in (4.7b) and $\frac{N}{p_{\max}^0}$ in (4.7c) normalize J_3 in relation to J_4. Both objectives intend to distribute the UAVs among the checkpoints and reduce the time between consecutive visits.

The four objectives are weighted and combined in the overall performance criterion

$$\min q_1 \cdot J_1 + q_2 \cdot J_2 + q_3 \cdot J_3 + q_4 \cdot J_4 \,. \tag{4.8}$$

4.5.4 Results

Simulation runs over a time period of 150 s were conducted for three different scenario versions: waypoints only, checkpoints only, and the full multi-objective scenario involving both waypoints and checkpoints. For each setup, 2 sensor-equipped UAVs moving within a $[-F, F] \times [-F, F] \times [0, F/2] \subset \mathbb{R}^3$ domain were considered. It is assumed that the simulation starts at time $t = 0$ while the puff release happened at some point in the past. Table 4.1 (page 85) summarizes all relevant simulation parameters. See Figure 4.8 for a snapshot of a simulation run.

The entire simulation involving two quadrotors and the corresponding recurring solver calls for waypoint generation and vehicle control, respectively, was run on one single machine (Intel®Core™i7-4790K CPU @ 4.00GHz, 16 GB RAM) with real-time factor 1.

By updating the control inputs every Δt s, the approximation error between the quadrotor model used in the MILP formulation and the more complex Gazebo model is compensated,

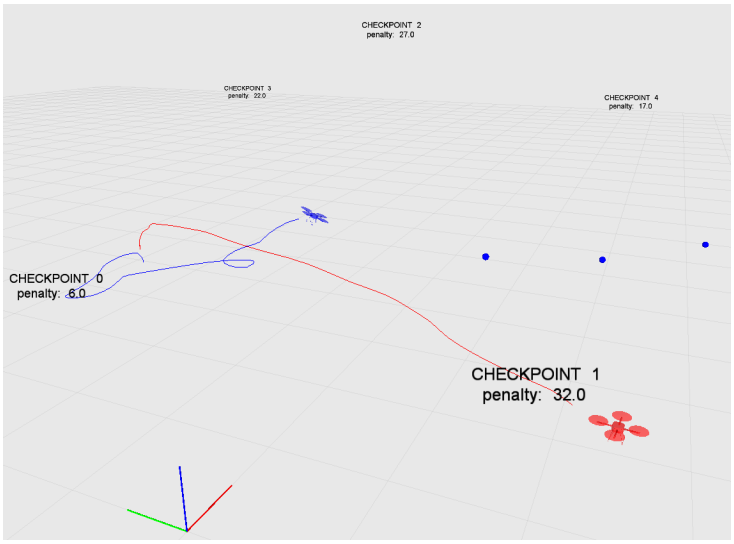


Figure 4.8: Snapshot of a simulation involving 2 quadrotors and 5 checkpoints visualized in RViz. The blue spheres represent the waypoints calculated for the blue quadrotor. The blue and red lines indicate the quadrotors' motion up to the time of the snapshot.

permitting the assumption that real quadrotor UAVs could also be controlled in that way. Moreover, it has to be considered that the UAVs do not take measurements at the exact waypoint locations. The MPC guides them such that distances to waypoints are minimized along with other possibly contrasting objectives. Hence, the measurements taken every Δt_m s can only be assumed close to the optimized locations.

After each measurement, a UAV requests its teammates' measurement data and based on the collective information updates the parameter estimate. For each scenario, the RMSE of the parameter estimate over time was evaluated in 10 simulation runs. The results are shown in Figure 4.9. As expected, the best error reduction is obtained in the waypoints-only scenario, where optimal parameter estimation is the only objective and the UAVs can concentrate on following their waypoints. In the checkpoints-only scenario, the UAVs focus on cooperative patrol, no waypoints are calculated. Still, measurements are taken every Δt_m s and the estimate is updated accordingly. That way, the RMSE is not reduced as much as in the waypoints-only case but still fairly well, possibly due to the checkpoints being conveniently located in the puff's dispersion domain. In the multi-objective scenario, the waypoints deviate the UAVs from heading for a checkpoint. Their trajectories are drawn to regions, where more valuable measurements can be gathered. This trade-off is reflected by the RMSE lying between the two single-objective scenarios at the end of the simulated time period. It can be expected that this effect would be more obvious if the checkpoints were located further away from the puff's center.

The conflict between the mission objectives is also reflected by the mean and maximum penalty values of the checkpoints depicted in Figure 4.10. As can be seen, the maximum penalty over time is significantly lower in the checkpoints-only scenario since the vehicles patrol the checkpoints more quickly.

In Figure 4.11, the convergence of the single parameters is plotted over the number of estimate updates. It shows that all parameters are estimated comparably well. However, some curves are monotonically decreasing, while others start with a large peak before converging to the true

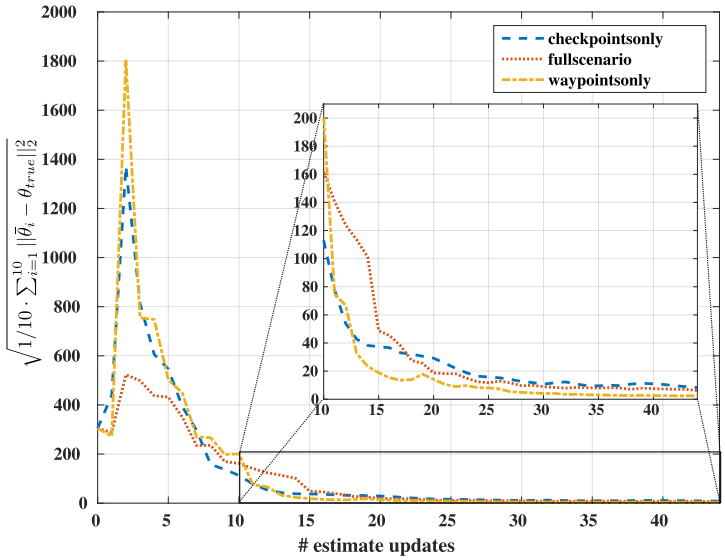


Figure 4.9: RMSE of the parameter estimate in 10 simulation runs for three different scenario versions: checkpoints only (blue dashed line), waypoints only (yellow dash dotted line), and both combined (red dotted line).

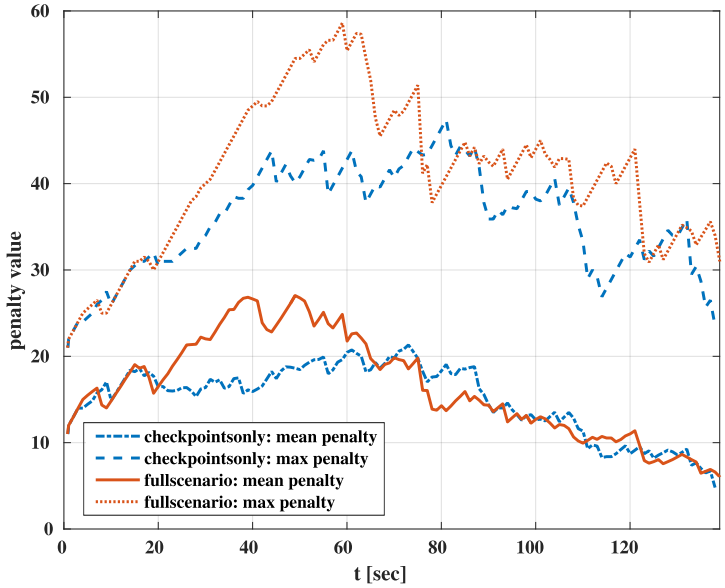


Figure 4.10: Mean of the maximum and mean checkpoint penalties in 10 simulation runs for the checkpoints-only and the full scenario.

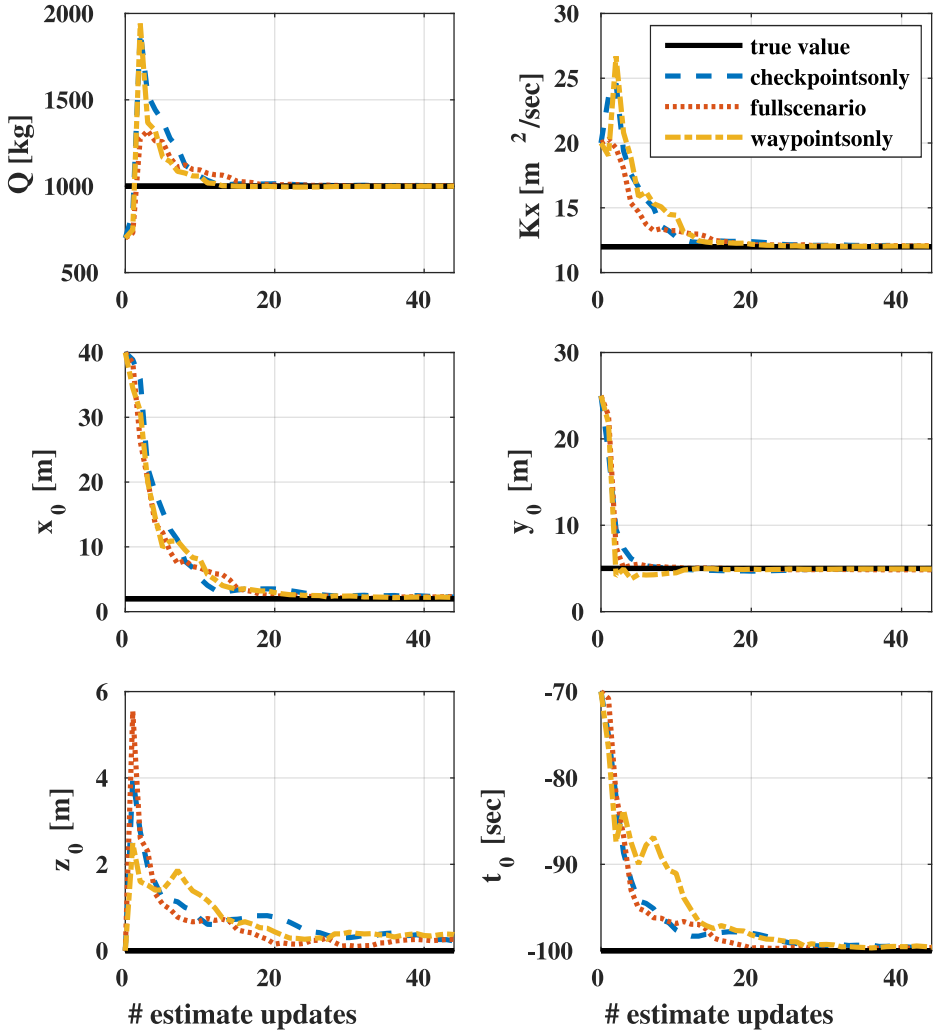


Figure 4.11: Mean convergence of the single parameters in 10 simulation runs for the three scenario versions.

parameter value. This peak appears accordingly in Figure 4.9 and is due to the low number and low quality of the first measurements taken near the unoptimized initial UAV positions.

parameter	value	unit
domain size	$F = 500$	[m]
initial UAV positions	$\mathbf{x}_0 = (20, 20, 10)$ $\mathbf{x}_1 = (25, 25, 10)$	[m]
maximum velocity	$v_{\max} = 8$	[m/s]
maximum acceleration	$u_{\max} = 3$	[m/s^2]
control interval	$\Delta t = 1$	[s]
prediction horizon	$N = 10$	
communication range	$d_{\text{com}} = 1000$	[m]
sensing interval	$\Delta t_m = 2$	[s]
waypt. sequence length	$n_w = 4$	
number of checkpoints	$n_C = 5$	
checkpoint locations and initial penalties $p_c(t_0)$	$(10, \ 20, \ 10, \ 20)$ $(10, \ -20, \ 10, \ 15)$ $(60, \ 10, \ 30, \ 10)$ $(70, \ 40, \ 20, \ 5)$ $(80, \ -20, \ 20, \ 0)$	[m, m, m, −]
penalty increase	$\Delta p = 1$	
measurement range	$d_{\text{check}} = 5$	[m]
objective weights	$q_1 = 2, q_2 = 1$ $q_3 = q_4 = 2000$	
true puff parameters	$\boldsymbol{\theta}_{\text{true}} = (Q, K_x, x_0, y_0, z_0, t_0)$ $= (1000, 12, 2, 5, 0, -100)$	([kg], [m^2/s], [m], [m], [m], [s])
initial estimate	$\bar{\boldsymbol{\theta}}_0 = (700, 20, 40, 25, 10^{-12}, -70)$	([kg], [m^2/s], [m], [m], [m], [s])
diffusivity parameter	$K_z = 0.2113$	[m^2/s]
wind speed	$v_{wx} = 0.5$	[m/s]

Table 4.1: Simulation parameters for the cooperative estimation and patrol scenario.

4.6 More Applications of Dynamic Data-Driven Cooperative Control

The proposed dynamic data-driven cooperative control approach can versatilely be applied to solve various other multi-objective monitoring tasks. Two examples, each highlighting distinctive problem features the proposed scheme is able to deal with, are briefly described below.

4.6.1 Sensor-Equipped Fixed-Wing UAVs Avoiding Collisions

An obvious modification of the application scenario detailed in Section 4.5 is the use of fixed-wing instead of quadrotor UAVs. However, the task combining waypoint following and checkpoint patrol as set up for the quadrotors could not be solved equally well by fixed-wing UAVs due to their completely different flight behavior characterized by permanent forward motion and large turning radii. These properties, though, make them suitable for sensing tasks in large-scale domains, e.g. when the mass of the released contaminant is much higher than assumed in the previous scenario or stronger winds and turbulences further spread the puff in the atmosphere.

The waypoint calculation approach can in such settings be applied to provide optimized sensing trajectories for a team of sensor-equipped fixed-wing UAVs while additional constraints affecting the UAVs' cooperative mobility, such as collision avoidance or connectivity maintenance, can be handled by mixed-integer MPC.

For MPC of airplane motion, the hybridized motion dynamics model described in Section 2.4.1 is applied. Hybridization and discretization provided the PWA model description (2.31), with matrices $\mathbf{A}_i^D$, $\mathbf{B}_i^D$ and vector $\mathbf{a}_i^D$ belonging to partition $\mathcal{S}_i$, $i = 1, 2, \ldots, 8$. In order to select the appropriate system approximation depending on the current value of the airplane's orientation φ^k at time step k of the prediction horizon, a set of binary variable $\delta_i^k \in \{0, 1\}$ is introduced such that

$$\delta_i^k = 1 \quad \Leftrightarrow \quad \varphi^k \in \mathcal{S}_i \,. \tag{4.9}$$

Since the partitions are disjoint, the mapping of φ^k to the corresponding $\mathcal{S}_i$ is unique, i.e.

$$\sum_{i=1}^{8} \delta_i^k = 1 \tag{4.10}$$

needs to hold at any time step k. Having selected the ith partition, the airplane's state vector $\mathbf{x}_{fw}^k$ can be updated according to the corresponding system dynamics by

$$\mathbf{x}_{fw}^{k+1} = \sum_{i=1}^{8} \mathbf{w}_i^k \,, \quad \text{where} \tag{4.11a}$$

$$\mathbf{w}_i^k = \left(\mathbf{A}_i^D \mathbf{x}_{fw}^k + \mathbf{B}_i^D \mathbf{u}_{fw}^k + \mathbf{a}_i^D \right) \cdot \delta_i^k \,. \tag{4.11b}$$

Due to the periodicity of sin and cos, the partitions $\mathcal{S}_i$ only cover the interval $[-\pi, \pi]$. However, φ^k might become greater than π or less than $-\pi$ within the prediction horizon N, thus

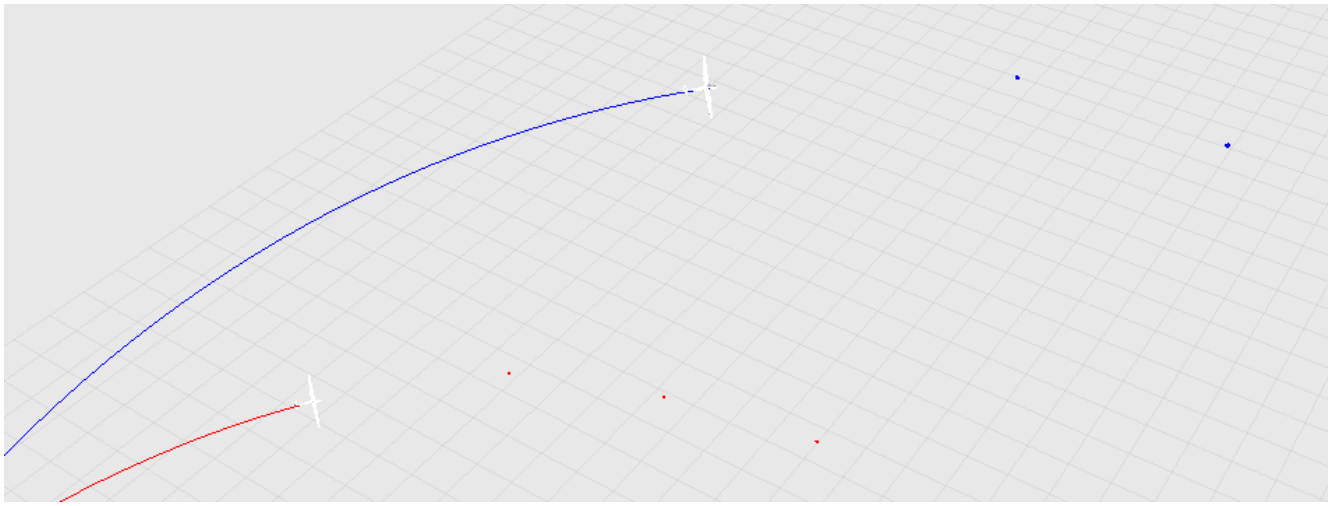


Figure 4.12: Simulation snapshot showing two fixed-wing UAVs following their individual optimized waypoints (blue and red spheres) while avoiding collisions.

transitions between $\mathcal{S}_1$ and $\mathcal{S}_8$ have to be enabled. This is done by introducing additional binary variables $\delta^k_{\varphi 1}$ and $\delta^k_{\varphi 2}$, such that

$$\delta^k_{\varphi 1} = 1 \quad \Leftrightarrow \quad \varphi^k \geq \pi \quad \text{and} \tag{4.12a}$$

$$\delta^k_{\varphi 2} = 1 \quad \Leftrightarrow \quad \varphi^k < -\pi\,, \tag{4.12b}$$

and by replacing all occurrences of φ^k by the expression $\varphi^k - \delta^k_{\varphi 1} \cdot 2\pi + \delta^k_{\varphi 2} \cdot 2\pi$.

As in previous applications, (4.9), (4.11), and (4.12) are transformed into mixed-integer linear inequalities using the Big-M method (cf. Section 2.3.2). Collision avoidance is modeled as described in Section 2.5.3.

A simulation snapshot of the considered application scenario is shown in Figure 4.12.

4.6.2 Quadrotor Communication Bridge for a Fixed-Wing UAV

Another example of a problem type that can be solved by the proposed dynamic data-driven sensing and control scheme involves heterogeneous UAVs that pursue different task objectives. This is in contrast to the previously described monitoring scenarios in Sections 4.5 and 4.6.1, where all vehicles have identical tasks. Now, the idea is to deploy a fixed-wing UAV following optimized waypoints for gathering data on an atmospheric dispersion process in a large-scale domain. At the same time, also a quadrotor UAV at much lower speed performs the same sensing task, but can thereby only cover a much smaller domain close to the ground. Likewise, an autonomous ground vehicle could be used.

In any case, a large distance between both sensor vehicles can be expected that prohibits direct communication. In order to still enable efficient cooperative sensing exploiting collectively gathered measurement data, multiple quadrotor UAVs are employed as routers forming a communication bridge between the sensors. For this purpose, they have to cooperatively take positions along the connecting line between the two sensor UAVs. In fact, the connecting line

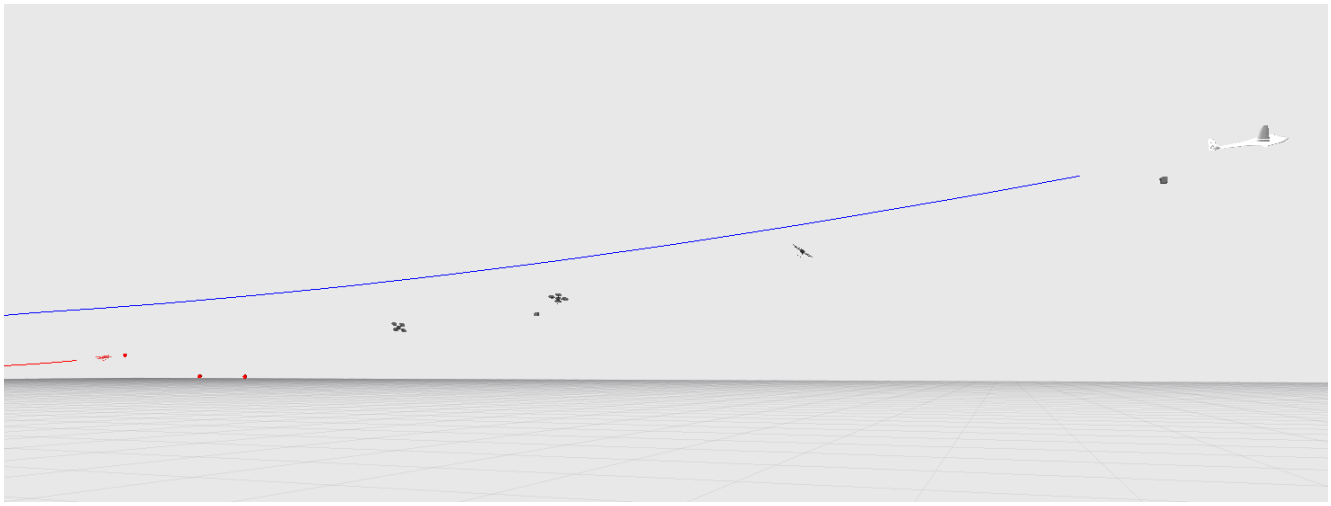


Figure 4.13: Simulation snapshot showing a fixed-wing UAV (blue trajectory) and a quadrotor UAV (red trajectory) collecting data along their individual optimized waypoints. Three additional (gray) quadrotors serve as communication bridge enabling information exchange between the sensor UAVs. Grey boxes mark their target positions.

between the last waypoint in each sensor vehicle's current waypoint sequence is used to determine evenly distributed target positions corresponding to the number of quadrotors available for routing. This is done by each router UAV individually, which requires local knowledge of the sensor UAVs' waypoints. As long as an intact connection to all other team members exists, this information can easily be shared. Otherwise, the last known waypoints or an estimate of their location is used.

The calculated target positions for the router UAVs serve as input for the mixed-integer MPC that determines appropriate vehicle control inputs implying optimal target assignment among the teammates. The cooperation logic for target assignment is modeled in the same way as for the target observation scenario presented in Section 2.5.

Thus, there are two types of responsibilities in the team's cooperative mission – sensor UAVs following optimized waypoints and router UAVs cooperatively occupying target positions to form a communication bridge. All vehicles can still be controlled using identical MPC implementations, i.e. one MILP representation of the overall task combining cooperative sensing and cooperative routing. Also, heterogeneity in the vehicles' motion dynamics is not a problem since each UAV in each control cycle reassembles the MILP of its local subsystem using the motion dynamics model corresponding to the type of its neighbor vehicle as mentioned in Section 4.4.

A simulation snapshot of the considered application scenario is shown in Figure 4.13.

4.7 Summary

A novel optimization-based dynamic data-driven cooperative control scheme has been derived by combining the mixed-integer MPC approach for cooperative multi-vehicle systems proposed in Chapter 2 with the sequential optimum design-based adaptive sensing approach for dynamic process estimation proposed in Chapter 3. It enables a permanent interaction of parameter

estimation and sensor motion for maximized effectiveness of the data collection process. In contrast to existing DDDAS solutions, the proposed scheme can handle additional cooperative mobility tasks and is, therefore, suitable for the control of multiple cooperating sensor vehicles in multi-objective monitoring applications.

Since already during the development of the components for cooperative control and adaptive sensing, respectively, special emphasis was put on (scalable) computational efficiency and decentralized applicability, these characteristics also apply to their combination and make the presented dynamic data-driven sensing and control loop flexible and scalable to arbitrary team sizes. It has been successfully evaluated in a representative simulated multi-objective monitoring scenario combining cooperative sensing and patrol (Section 4.5), but can also be adapted to various other application scenarios, two examples of which are outlined in Section 4.6.

For the evaluation, a simulation framework (Section 4.4) has been developed based on ROS and Gazebo, both representing widely used state-of-the-art robot software libraries and tools. This enables testing under realistic conditions and will simplify the transfer of the implemented software from simulated to real robots.

5 Conclusions

5.1 Summary of Contributions

This thesis addresses the challenging problem of controlling a team of Unmanned Aerial Vehicles (UAVs) in order to cooperatively monitor the environment and identify an approximate model of an atmospheric dispersion process in a most optimal and real-time efficient manner. Within this scope, three main contributions to the field of UAV applications in cooperative monitoring scenarios were made, which are reviewed in this section.

Decentralized Model-Predictive Cooperative Control based on MILP

An approach for efficient optimization-based control of multiple cooperating unmanned vehicles has been developed (Chapter 2). It is based on discrete-time linear approximations of the differential equations describing the vehicles' motion dynamics as well as mixed logical constraints representing the cooperative interaction of the team members. Formulating the cooperative mission as a Mixed Integer Linear Program (MILP) enables the use of highly efficient mathematical solvers, and employing a Receding Horizon Control (RHC) scheme further reduces the computational effort that is required to obtain locally optimal vehicle control inputs. These control inputs implicitly reflect the vehicles' assignment to different tasks and, hence, their cooperation for reaching the common mission objective.

By decomposing the overall time horizon into smaller time windows and repeatedly solving MILPs in a model-predictive feedback control fashion, the obtained final state does not reach the same optimality level as that obtained from a single feedforward MILP solution. However, the cooperative control problem can be solved much more efficiently and at the same time be continuously adapted to the true system state. This is even more advantageous in dynamically changing environments as targeted in this thesis, where feedforward approaches are not applicable at all.

Much more importantly, decentralized mixed-integer MPC of local subsystems significantly reduces the overall computation time compared to solving a single centralized MILP, especially for larger numbers of involved vehicles. The proposed decentralization (Section 2.4.3) even moves one step further and provides an upper bound for the computational effort of each individual controller and, hence, makes the approach scalable to arbitrary team sizes while maintaining real-time efficiency.

In summary, the developed approach bridges the gap between computationally expensive feedforward optimal control solutions and realtime-efficient heuristic feedback control approaches.

Applications of the proposed cooperative controller in a number of different multi-vehicle scenarios (Sections 2.5, 4.5, and 4.6) prove its versatility and extensibility.

Sequential Optimum Design for Adaptive Vehicle-Specific Sensor Movement

For solving the problem of where to send sensor-equipped UAVs in order to obtain the most valuable measurements, an optimization-based approach for the calculation of vehicle-specific sensing trajectories was proposed that employs the concept of Optimum Experimental Design (OED) (Chapter 3). Purpose of the cooperative data collection is to obtain a more accurate estimate of the parameters of an atmospheric dispersion process model by repeatedly solving a weighted Nonlinear Least Squares (wNLLSQ) problem. As for the cooperative controller, emphasis was put on computational efficiency of the procedure enabling decentralized application on board a UAV with limited computing capacity. For this reason, the basic Gaussian puff solution was employed for modeling an atmospheric dispersion process instead of a more complex and computationally much more expensive process model (Section 3.3.1).

By optimum design of sequences of waypoints for each individual UAV, fewer measurements are required for accurate parameter estimation, making the measurement process more effective and more efficient. Moreover, an optimum design problem can allow for constrained design spaces, which in the considered use case corresponds to the vehicles' motion capabilities that can, thus, be optimally exploited.

Since optimum designs for nonlinear models depend on the parameters to be estimated ("chicken-and-egg" problem), a sequential design procedure was employed (Section 3.4.4). In every iteration, the current parameter estimate is updated incorporating newly collected measurements and new design points in the form of optimized waypoints are added to the overall optimum design. That way, the sensing process continuously adapts to the collected data. The sequential nature of the proposed method also enables the adaptation to the sensor UAV's true motion by accordingly selecting the intial conditions for the waypoint optimization.

Comprehensive simulation results illustrate the adaptive sensing approach from different perspectives and prove its effectiveness in comparison to different non-adaptive sensor motion patterns and even stationary sensor networks (Section 3.5). A case study for a comparison of different optimality criteria identified D-optimality to best suit the considered application and a number of four waypoints per sequence update provided the best tradeoff between computational effort and estimation error reduction.

DDDAS for Cooperating UAVs in Multi-Objective Monitoring Scenarios

In order to obtain an efficient decentralized dynamic data-driven sensing and control scheme for cooperating UAVs in multi-objective monitoring tasks (Chapter 4), the mixed-integer MPC approach (Chapter 2) was coupled to the optimum design of sensing trajectories (Chapter 3) in a repeating feedback loop (Figure 4.1). By regularly updating the optimized sequence of waypoints depending on the gathered sensor data and the current vehicle state, the sensing trajectories permanently adapt to the UAVs' behavior and, at any time, represent the currently best possible sensor movement.

This adaptation is essential since the mixed-integer MPC approach is able and meant to deal with multiple mission objectives, the balancing of which is represented by the MILP cost function. The evaluation of the proposed Dynamic Data-Driven Application System (DDDAS) in a simulated representative multi-objective monitoring scenario (Section 4.5) verifies its performance in solving cooperative mobility tasks in combination with adaptive sensing for optimal parameter estimation. Simulations were performed using a ROS framework and realistic Gazebo

simulation models of quadrotor and fixed-wing UAVs (Section 4.4), which poses a necessary and reasonable intermediate step towards experimenting with real hardware.

Enhancing a data-driven control system for adaptive sensing with the ability to simultaneously perform additional vehicle cooperation tasks is novel in the field of DDDAS. However, this combination is a typical feature of practical problems in disaster response or environmental monitoring, as can , i.a., be seen from the examples in Section 4.6. Therefore, the proposed scheme may serve as a starting point for useful applications of real multi-UAV systems in those scenarios, especially since it was specifically designed for decentralized use at limited computational resources.

5.2 Outlook

The developed dynamic data-driven sensing and control scheme for cooperative multi-vehicle systems offers various starting points for extension and further research questions. Beside obvious potentials for modification, such as involving other vehicle types (e.g. autonomous ground vehicles) or considering more complex process models (e.g. with multiple sources or splitting plumes), it remains an interesting question how communication loss or delays would affect the vehicles' cooperative performance. Also, the combination of cooperating mobile sensors with a static sensor network, e.g. as in [76, 152, 21], poses an extension worth considering and could be realized at manageable effort. Identifying and implementing an approach for decentralized data fusion (cf. [123]) would complete the proposed solution.

During the setup of simulation experiments, it turned out that the quality of the parameter estimation process strongly depends on the choice of the initial measurement locations. From a more theoretical point of view, an (optimization-based) a priori calculation of the UAVs' initial deployment would, therefore, be useful. It could further improve and accelerate the data collection, especially since a bad (initial) estimate, in turn, negatively affects the waypoint design.

Regarding applications of the proposed DDDAS in real-world scenarios, it has to be investigated whether or not different simplifying assumption may be kept up or have to be replaced. This includes, e.g., the basic motion dynamics models for quadrotor and fixed-wing UAVs that might not suffice to accurately control real UAVs at challenging wind conditions. Extending the models to account for wind influence can be a way to deal with this risk. In terms of the employed sensor model, a modification for handling correlated measurements should be considered. As stated in [155], the assumption of measurement noise being zero-mean, white Gaussian and spatially uncorrelated is often made, but not realistic, and, therefore, might lead to weaker performance in real-world applications compared to the simulated scenarios. On the other hand, more complex models lead to an increase in computational effort and a prior evaluation of the UAVs' actual onboard processing capacities is recommended.

Bibliography

[1] AERMOD website https://www3.epa.gov/ttn/scram/dispersion_prefrec.htm#aermod. Accessed on September 22, 2016.

[2] CALPUFF website http://www.src.com/calpuff/calpuff1.htm. Accessed on September 22, 2016.

[3] Gazebo Tutorial: LiftDragPlugin, http://gazebosim.org/tutorials?tut=aerodynamics&cat=plugins. Accessed December 17, 2016.

[4] Image from website http://bestairplanephotos2013.blogspot.de/2013/07/airplane-parts.html. Accessed on December 20, 2016.

[5] Image from website http://www.mopo.de/hamburg/chemie-unfall-billbrook--so-giftig-war-diese-wolke-22375172. Accessed on April 29, 2016.

[6] Image from website http://www.morgenpost.de/vermischtes/article205568655/Chemiefabrik-brennt-hoechste-Alarmstufe-ausgerufen.html. Accessed on April 29, 2016.

[7] Image from website http://www.telegraph.co.uk/news/worldnews/northamerica/usa/12071857/Infrared-camera-reveals-huge-methane-gas-leak-wreaking-havoc-in-Los-Angeles.html. Accessed on April 29, 2016.

[8] Image from website http://www.welt.de/vermischtes/article137388776/Riesige-Giftwolke-zieht-ueber-Spanien.html. Accessed on April 29, 2016.

[9] A. Khan, B. Rinner, and A. Cavallaro. Cooperative Robots to Observe Moving Targets: Review. *IEEE Transactions on Cybernetics*, PP(99):1–12, 2016.

[10] S. M. Adams and C. J. Friedland. A Survey of Unmanned Aerial Vehicle (UAV) Usage for Imagery Collection in Disaster Research and Management. *9th International Workshop on Remote Sensing for Disaster Response*, 2011.

[11] A. Atkinson, A. N. Donev, and R. D. Tobias. *Optimum Experimental Designs, With SAS*. Oxford University Press, 2007.

[12] A. Bemporad, F. Borrelli, and M. Morari. Optimal controllers for hybrid systems: stability and piecewise linear explicit form. In *Proceedings of the 39th IEEE Conference on Decision and Control (CDC)*, volume 2, pages 1810–1815, 2000.

[13] A. Bemporad, F. Borrelli, and M. Morari. Model Predictive Control Based on Linear Programming - The Explicit Solution. 47(12):1974–1985, 2002.

[14] A. Bemporad and G. Ferrari-Trecate. Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control*, 45(10):1864–1876, 2000.

[15] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–427, 1999.

[16] J. Binney, A. Krause, and G. S. Sukhatme. Optimizing waypoints for monitoring spatiotemporal phenomena. *The International Journal of Robotics Research*, 32(8):873–888,

2013.

[17] P. Boccardo, F. Chiabrando, F. Dutto, F. Tonolo, and A. Lingua. UAV Deployment Exercise for Mapping Purposes: Evaluation of Emergency Response Applications. *Sensors*, 15(7):15717–15737, 2015.

[18] F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*. Springer, 2003.

[19] R. D. Braatz. Cooperative Control [About This Issue]. *IEEE Control Systems*, 34(4):8–11, 2014.

[20] C. Branca and R. Fierro. Hierarchical Optimization Strategies for Deployment of Mobile Robots. *International Journal of Intelligent Control Systems*, 11(3):141–153, 2006.

[21] J. Brink. Boundary tracking and estimation of pollutant plumes with a mobile sensor in a low-density static sensor network. *Urban Climate*, 14:383–395, 2015.

[22] D. Buske, M. T. Vilhena, T. Tirabassi, and B. Bodmann. Air Pollution Steady-State Advection-Diffusion Equation: The General Three-Dimensional Solution. *Journal of Environmental Protection*, 03(09):1124–1134, 2012.

[23] M. Buss, O. von Stryk, R. Burlisch, and G. Schmidt. Towards Hybrid Optimal Control. *at-Automatisierungstechnik*, 48(9):448–459, 2000.

[24] G. Cabrita and L. Marques. Estimation of Gaussian Plume Model Parameters Using the Simulated Annealing Algorithm. In M. A. Armada, A. Sanfeliu, and M. Ferre, editors, *ROBOT2013: First Iberian Robotics Conference: Advances in Robotics*, volume 2, pages 369–380. 2014.

[25] J. A. Caley and G. A. Hollinger. Data-driven Comparison of Spatio-temporal Monitoring Techniques. In *Proceedings of the IEEE/MTS OCEANS Conference*, 2015.

[26] C. J. Cannell, A. S. Gadre, and D. J. Stilwell. Boundary Tracking and Rapid Mapping of A Thermal Plume Using an Autonomous Vehicle. In *Proceedings of the IEEE/MTS OCEANS Conference*, pages 1–6, 2006.

[27] C. J. Cannell and D. J. Stilwell. A comparison of two approaches for adaptive sampling of environmental processes using autonomous underwater vehicles. In *Proceedings of the MTS/IEEE OCEANS Conference*, pages 1514–1521, 2005.

[28] C. G. Cassandras and W. Li. Sensor Networks and Cooperative Control. In *Proceedings of the 44th IEEE Conference on Decision and Control (CDC) and European Control Conference (ECC)*, pages 4237–4238, 2005.

[29] G. C. Chasparis and J. S. Shamma. Linear-programming-based multi-vehicle path planning with adversaries. In *Proceedings of the American Control Conference (ACC)*, pages 1072–1077, 2005.

[30] H. L. Choi and J. P. How. Continuous trajectory planning of mobile sensors for informative forecasting. *Automatica*, 46:1266–1275, 2010.

[31] H.-L. Choi, J. P. How, and J. A. Hansen. Ensemble-Based Adaptive Targeting of Mobile Sensor Networks. In *Proceedings of the American Control Conference (ACC)*, pages 2393–2398, 2007.

[32] F. J. Christophersen. *Optimal Control and Analysis for Constrained Piecewise Affine Systems*. PhD thesis, Swiss Federal Institute of Technology (ETH) Zurich, 2006.

[33] V. N. Christopoulos and S. Roumeliotis. Adaptive Sensing for Instantaneous Gas Release Parameter Estimation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4450–4456, 2005.

[34] V. N. Christopoulos and S. Roumeliotis. Multi Robot Trajectory Generation for Single Source Explosion Parameter Estimation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2803–2809, 2005.

[35] J. Clark and R. Fierro. Cooperative hybrid control of robotic sensors for perimeter detection and tracking. In *Proceedings of the American Control Conference (ACC)*, pages 3500 – 3505 vol. 5, 2005.

[36] C. Coopmans, B. Stark, A. Jensen, Y. Chen, and M. McKee. Cyber-Physical Systems Enabled by Small Unmanned Aerial Vehicles. In K. P. Valavanis and G. J. Vachtsevanos, editors, *Handbook of Unmanned Aerial Vehicles*, chapter 117, pages 2835–2860. Springer Netherlands, 2015.

[37] K. Daniel, B. Dusza, A. Lewandowski, and C. Wietfeld. AirShield: A system-of-systems MUAV remote sensing architecture for disaster response. In *Proceedings of the 3rd Annual IEEE International Systems Conference*, pages 196–200, 2009.

[38] K. Daniel, S. Rohde, N. Goddemeier, and C. Wietfeld. A communication aware steering strategy avoiding self-separation of flying robot swarms. In *Proceedings of the IEEE International Conference on Intelligent Systems*, pages 254–259, 2010.

[39] F. Darema. Dynamic data driven applications systems: A new paradigm for application simulations and measurements. In *4th International Conference on Computational Science*, volume 3038 of *Lecture Notes in Computer Science*, pages 662–669. Springer, 2004.

[40] B. De Schutter, W. P. M. H. Heemels, J. Lunze, and C. Prieur. Survey of modeling, analysis, and control of hybrid systems. In *Handbook of Hybrid Systems Control – Theory, Tools, Applications*, chapter 2, pages 31–56. 2009.

[41] M. Demetriou and D. Uciński. State Estimation of Spatially Distributed Processes Using Mobile Sensing Agents. In *Proceedings of the American Control Conference (ACC)*, pages 1770–1776. IEEE, 2011.

[42] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-Based Multirobot Coordination: A Survey and Analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006.

[43] Y. Ding, M. Zhu, Y. He, and J. Jiang. P-CMOMMT algorithm for the cooperative multirobot observation of multiple moving targets. *Proceedings of the World Congress on Intelligent Control and Automation*, 2:9267–9271, 2006.

[44] M. Dunbabin and L. Marques. Robotics for Environmental Monitoring: Significant Advancements and Applications. *IEEE Robotics & Automation Magazine*, 19(1):24–39, 2012.

[45] M. G. Earl and R. D'Andrea. Multi-Vehicle Cooperative Control Using Mixed Integer Linear Programming. In *Cooperative Control of Distributed Multi-Agent Systems*, pages 231–259. John Wiley & Sons, Ltd, 2007.

[46] J. Euler, A. Horn, D. Haumann, J. Adamy, and O. von Stryk. Cooperative N-Boundary Tracking in Large Scale Environments. In *Proceedings of the IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems*, 2012.

[47] J. Euler, T. Ritter, S. Ulbrich, and O. von Stryk. Centralized Ensemble-Based Trajectory Planning of Cooperating Sensors for Estimating Atmospheric Dispersion Processes. In S. Ravela and A. Sandu, editors, *Dynamic Data-Driven Environmental Systems Science*, volume 8964 of *Lecture Notes in Computer Science*. Springer International Publishing, 1 edition, 2015.

[48] J. Euler and O. von Stryk. Decentralized Data-Driven Control of Cooperating Sensor-Carrying UAVs in a Multi-Objective Monitoring Scenario. *IFAC-PapersOnLine*, 50(1):15828–15834, 2017.

[49] J. Euler and O. von Stryk. Optimized Vehicle-Specific Trajectories for Cooperative Process Estimation by Sensor-Equipped UAVs. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3397–3403, 2017.

[50] C. A. F. Ezequiel, M. Cua, N. C. Libatique, G. L. Tangonan, R. Alampay, R. T. Labuguen, C. M. Favila, J. L. E. Honrado, V. Canos, C. Devaney, A. B. Loreto, J. Bacusmo, and B. Palma. UAV aerial imaging applications for post-disaster assessment, environmental management and infrastructure development. *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 274–283, 2014.

[51] E. Franco, L. Magni, T. Parisini, M. M. Polycarpou, and D. M. Raimondo. Cooperative Constrained Control of Distributed Agents with Nonlinear Dynamics and Delayed Information Exchange: A Stabilizing Receding-Horizon Approach. *IEEE Transactions on Automatic Control*, 53(1):324–338, 2008.

[52] P. E. Gill, E. Wong, W. Murray, and M. A. Saunders. User's Guide for SNOPT Version 7.5. Technical report, Center for Computational Mathematics, University of California, San Diego, 2015.

[53] M. Glocker, C. Reinl, and O. von Stryk. Optimal Task Allocation and Dynamic Trajectory Planning for Multi-Vehicle Systems using Nonlinear Hybrid Optimal Control. In *Proceedings of the 1st IFAC Workshop on Multivehicle Systems*, pages 38–43, 2006.

[54] R. Goebel, R. Sanfelice, and A. Teel. Hybrid dynamical systems. *IEEE Control Systems*, 29(2):28–93, 2009.

[55] R. Goebel, R. Sanfelice, and A. Teel. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, 2012.

[56] R. Graham and J. Cortés. Cooperative adaptive sampling via approximate entropy maximization. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2009.

[57] I. E. Grossmann and S. Lee. Generalized Convex Disjunctive Programming: Nonlinear Convex Hull Relaxation. *Computational Optimization and Applications*, 26:83–100, 2003.

[58] Gurobi Optimization Inc. Gurobi 6.5 Performance Benchmarks, 2015.

[59] Gurobi Optimization Inc. Gurobi Optimizer Reference Manual, 2016.

[60] P. C. Hansen, V. Pereyra, and G. Scherer. *Least Squares Data Fitting with Applications*. The Johns Hopkins University Press, 2013.

[61] D. J. Harvey, T.-F. Lu, and M. A. Keller. Comparing Insect-Inspired Chemical Plume Tracking Algorithms Using a Mobile Robot. *IEEE Transactions on Robotics*, 24(2):307–317, 2008.

[62] J. Haugen and L. Imsland. Monitoring an Advection-Diffusion Process Using Aerial Mobile Sensors. *Unmanned Systems*, 3(3):221–238, 2015.

[63] M. J. Hirsch and D. Schroeder. On the Decentralized Cooperative Control of Multiple Autonomous Vehicles. In K.P. Valavanis and G. Vachtsevanos, editors, *Handbook of Unmanned Aerial Vehicles*, chapter 64, pages 1577–1600. Springer Netherlands, 2015.

[64] T. E. Hodrus. *Prozessführungsstrategien für hybride Systeme*. PhD Thesis, Universität Karlsruhe (TH), 2007.

[65] C. Hsieh, D. Marthaler, B. Nguyen, D. Tung, A. Bertozzi, and R. Murray. Experimental validation of an algorithm for cooperative boundary tracking. In *Proceedings of the American Control Conference (ACC)*, pages 1078–1083, 2005.

[66] S. M. Huck, P. Hokayem, D. Chatterjee, and J. Lygeros. Stochastic localization of sources using autonomous underwater vehicles. In *Proceedings of the American Control Conference (ACC)*, pages 4192–4197, 2012.

[67] ILOG. ILOG CPLEX 11.0 User's Manual, 2007.

[68] ILOG. *IBM ILOG CPLEX Optimization Studio: CPLEX 12.6 User's Manual*, 2015.

[69] H. Ishida, G. Nakayama, T. Nakamoto, and T. Moriizumi. Controlling a gas/odor plume-tracking robot based on transient responses of gas sensors. *IEEE Sensors Journal*, 5(3):537–545, 2005.

[70] M. Jaward, D. Bull, and N. Canagarajah. Sequential Monte Carlo methods for contour tracking of contaminant clouds. *Signal Processing*, 90(1):249–260, jan 2010.

[71] Z. Jin and A. L. Bertozzi. Environmental boundary tracking and estimation using multiple autonomous vehicles. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, pages 4918–4923, 2007.

[72] S. Karaman, E. Koyuncu, and G. Inalhan. Innovative Collaborative Task Allocation for UAVs. In P. K. Valavanis and J. G. Vachtsevanos, editors, *Handbook of Unmanned Aerial Vehicles*, pages 1601–1617. Springer Netherlands, 2015.

[73] P. Kathirgamanathan, R. Mckibbin, and R. McLachlan. Source Term Estimation of Pollution from an Instantaneous Point Source. *MODSIM*, 6:59–67, 2002.

[74] A. Khamis, A. Hussein, and A. Elmogy. Multi-robot Task Allocation: A Review of the State-of-the-Art. In A. Koubâa and J. R. Martínez-de Dios, editors, *Cooperative Robots and Sensor Networks*, pages 31–51. 2015.

[75] A. Khan, B. Rinner, and A. Cavallaro. Multiscale Observation of Multiple Moving Targets using Micro Aerial Vehicles. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 4642–4649, 2015.

[76] A. Khelil, C. Reinl, B. Ayari, F. K. Shaikh, P. Szczytowski, A. Ali, and N. Suri. Wireless Sensor Cooperation for a Sustainable Quality of Information. In *Pervasive Computing and Networking*, pages 71–100. 2011.

[77] N. Koenig and A. Howard. Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[78] J. Kuhn. *Model-Predictive Control of Cooperative Multi-Vehicle Systems Based on Discrete-Time Linear Systems*. Diploma Thesis, Technische Universität Darmstadt, 2009.

[79] J. Kuhn, C. Reinl, and O. von Stryk. Predictive Control for Multi-Robot Observation of Multiple Moving Targets Based on Discrete-Continuous Linear Models. *IFAC Proceedings Volumes*, 44(1):257–262, 2011.

[80] M. Kvasnica. *Efficient Software Tools for Control and Analysis of Hybrid Systems*. PhD thesis, Swiss Federal Institue of Technology (ETH) Zurich, 2008.

[81] M. Kvasnica, P. Grieder, M. Baotić, and F. J. Christophersen. *Multi-Parametric Toolbox (MPT)*. Swiss Federal Institute of Technology (ETH) Zurich, 2006.

[82] S. M. LaValle. *Planning Algorithms*. 2006.

[83] W. Li and C. G. Cassandras. A Cooperative receding horizon controller for multivehicle uncertain environments. *IEEE Transactions on Automatic Control*, 51(2):242–257, 2006.

[84] W. Li and C. G. Cassandras. Centralized and distributed cooperative Receding Horizon control of autonomous vehicle missions. *Mathematical and Computer Modelling*, 43(9-10):1208–1228, 2006.

[85] W. Li and J. E. Sutton. Optimization of Source Identification Algorithm Derived from Moth-Inspired Plume Tracing Strategies. In *Proceedings of the International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 79–84, 2007.

[86] S. Luke, K. Sullivan, L. Panait, and G. Balan. Tunably decentralized algorithms for cooperative target observation. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2005.

[87] J. Lunze. *Regelungstechnik 2*. 2004.

[88] R. Madankan, P. Singla, and T. Singh. Optimal information collection for source parameter estimation of atmospheric release phenomenon. In *Proceedings of the American Control Conference (ACC)*, pages 604–609, 2014.

[89] S. Markov and S. Carpin. A cooperative distributed approach to target motion control in multirobot observation of multiple targets. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[90] I. Maza, A. Ollero, E. Casado, and D. Scarlatti. Classification of Multi-UAV Architectures. In K. Valavanis and G. Vachtsevanos, editors, *Handbook of Unmanned Aerial Vehicles*, chapter 38, pages 954–975. Springer Netherlands, 2015.

[91] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk. Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo. In I. Noda, N. Ando, D. Brugali, and J. J. Kuffner, editors, *Simulation, Modeling, and Programming for Autonomous Robots*, volume 7628 of *Lecture Notes in Computer Science*, pages 400–411. Springer Berlin Heidelberg, 2012.

[92] S. Moon, D. H. Shim, and E. Oh. Cooperative Task Assignment and Path Planning for Multiple UAVs. In P. K. Valavanis and J. G. Vachtsevanos, editors, *Handbook of Unmanned Aerial Vehicles*, pages 1547–1576. Springer Netherlands, 2015.

[93] R. M. Murray. Recent Research in Cooperative Control of Multi-Vehicle Systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 129:571–583, 2007.

[94] R. Negenborn and J. Maestre. Distributed Model Predictive Control: An Overview and Roadmap of Future Research Opportunities. *IEEE Control Systems*, 34:87–97, 2014.

[95] P. P. Neumann, S. Asadi, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller. Autonomous Gas-Sensitive Microdrone: Wind Vector Estimation and Gas Distribution Mapping. *IEEE Robotics & Automation Magazine*, 19(1):50–61, 2012.

[96] L. V. Nguyen, S. Kodagoda, R. Ranasinghe, and G. Dissanayake. Information-driven Adaptive Sampling Strategy for Mobile Robotic Wireless Sensor Network. *IEEE Transactions on Control Systems Technology*, 24(1):372–379, 2016.

[97] J. Nocedal and S. J. Wright. *Numerical optimization*. 2006.

[98] P. Ögren, E. Fiorelli, and N. E. Leonard. Cooperative Control of Mobile Sensor Networks: Adaptive Gradient Climbing in a Distributed Environment. *IEEE Transactions on Automatic Control*, 49(8):1292–1302, 2004.

[99] H. Oh, H.-S. Shin, S. Kim, A. Tsourdos, and B. A. White. Cooperative Mission and Path Planning for a Team of UAVs. In P. K. Valavanis and J. G. Vachtsevanos, editors, *Handbook of Unmanned Aerial Vehicles*, chapter 62, pages 1509–1545. Springer Netherlands, 2015.

[100] A. Ollero, S. Lacroix, L. Merino, J. Gancet, J. Wiklund, V. Remuss, I. V. Perez, L. G. Gutierrez, D. X. Viegas, M. A. G. Benitez, A. Mallet, R. Alami, R. Chatila, G. Hommel, F. J. C. Lechuga, B. C. Arrue, J. Ferruz, J. R. Martinez-De Dios, and F. Caballero. Multiple eyes in the skies: architecture and perception issues in the COMETS unmanned air vehicles project. *IEEE Robotics and Automation Magazine*, 12(2):46–57, 2005.

[101] O. A. A. Orqueda and R. Fierro. Model-Predictive Path-Space Iteration for Multi-Robot Coordination. In D. Grundel, R. Murphey, P. Pardalos, and O. Prokopyev, editors, *Cooperative Systems: Control and Optimization*, pages 229–253. Springer, 2007.

[102] G. Pajares. Overview and Current Status of Remote Sensing Applications Based on Unmanned Aerial Vehicles (UAVs). *Photogrammetric Engineering & Remote Sensing*, 81(4):281–329, 2015.

[103] L. Panait and S. Luke. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems*, 11:387–434, 2005.

[104] L. E. Parker. ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

[105] L. E. Parker. Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets. *Autonomous Robots*, 12:231–255, 2002.

[106] L. E. Parker and B. A. Emmons. Cooperative Multi-Robot Observation of Multiple Moving Targets. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, number 3, pages 2082–2089, 1997.

[107] L. E. Parker and C. Touzet. Multi-Robot Learning in a Cooperative Observation Task. In L. E. Parker, G. Bekey, and J. Barhen, editors, *Distributed Autonomous Robotic Systems 4*, pages 391–401. Springer, 2000.

[108] L. Peng and K. Mohseni. Sensor driven feedback for puff estimation using unmanned aerial vehicles. In *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 562–569, 2014.

[109] L. Peng, M. Silic, R. O. Donnell, and K. Mohseni. A DDDAS Plume Monitoring System with Reduced Kalman Filter. *Procedia Computer Science*, 51:1–10, 2015.

[110] D. Pieri and J. A. Diaz. In Situ Sampling of Volcanic Emissions with a UAV Sensorweb: Progress and Plans. In *Dynamic Data-Driven Environmental Systems Science - First International Conference, DyDESS 2014, Cambridge, MA, USA, November 5-7, 2014, Revised Selected Papers*, volume LNCS 8964, pages 16–27. 2015.

[111] R. Pöllänen, H. Toivonen, K. Peräjärvi, T. Karhunen, T. Ilander, J. Lehtinen, K. Rintala, T. Katajainen, J. Niemelä, and M. Juusela. Radiation surveillance using an unmanned aerial vehicle. *Applied Radiation and Isotopes*, 67(2):340–344, 2009.

[112] S. S. Ponda, L. B. Johnson, A. Geramifard, and J. P. How. Cooperative Mission Planning for Multi-UAV Teams. In P. K. Valavanis and J. G. Vachtsevanos, editors, *Handbook of Unmanned Aerial Vehicles*, chapter 60, pages 1–3022. Springer Netherlands, 2015.

[113] D. Portugal, C. Pippin, R. P. Rocha, and H. Christensen. Finding optimal routes for multi-robot patrolling in generic graphs. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 363–369, 2014.

[114] D. Portugal and R. P. Rocha. Distributed Multi-Robot Patrol: A Scalable and Fault-Tolerant Framework. *Robotics and Autonomous Systems*, 61(12):1572–1587, 2013.

[115] D. Portugal and R. P. Rocha. Cooperative Multi-Robot Patrol with Bayesian Learning. *Autonomous Robots*, 40(5):929–953, 2016.

[116] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.

[117] E. Rafajłowicz. Optimum choice of moving sensor trajectories for distributed-parameter system identification. *International Journal of Control*, 43(5):1441–1451, 1986.

[118] M. Rahimi, M. Hansen, W. J. Kaiser, G. S. Sukhatme, and D. Estrin. Adaptive sampling for environmental field estimation using robotic sensors. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3692–3698, 2005.

[119] C. Reinl. *Trajektorien- und Aufgabenplanung kooperierender Fahrzeuge: Diskret-kontinuierliche Modellierung und Optimierung (Trajectory and Task Planning for Cooperating Vehicles: Discrete-Continuous Modeling and Optimization)*. PhD Thesis, Technische Universität Darmstadt, 2010.

[120] C. Reinl and O. von Stryk. Optimal Control of Cooperative Multi-Robot Systems using Mixed-Integer Linear Programming. In *Workshop on Robotics and Mathematics (RoboMat)*, 2007.

[121] A. Richards and J. How. Mixed-integer programming for control. In *Proceedings of the American Control Conference (ACC)*, pages 2676–2683, 2005.

[122] A. Richards and J. P. How. Robust distributed model predictive control. *International Journal of Control*, 80(9):1517–1531, 2007.

[123] M. F. Ridley, B. Upcroft, and S. Sukkarieh. Data Fusion and Tracking with Multiple UAVs. In P. K. Valavanis and J. G. Vachtsevanos, editors, *Handbook of Unmanned Aerial Vehicles*, pages 461–490. Springer Netherlands, 2015.

[124] M. Risler. *Behavior Control for Single and Multiple Autonomous Agents Based on Hierarchical Finite State Machines*. Fortschritt-Berichte VDI, Reihe 10: Informatik/Kommunikation, Technische Universität Darmstadt, 2009.

[125] T. Ritter. *PDE-Based Dynamic Data-Driven Monitoring of Atmospheric Dispersion Processes*. Phd thesis, Technische Universität Darmstadt, 2017.

[126] T. Ritter, J. Euler, S. Ulbrich, and O. von Stryk. Adaptive Observation Strategy for Dispersion Process Estimation Using Cooperating Mobile Sensors. *IFAC Proceedings Volumes*, 47(3):5302–5308, 2014.

[127] T. Ritter, J. Euler, S. Ulbrich, and O. von Stryk. Decentralized Dynamic Data-Driven Monitoring of Atmospheric Dispersion Processes. *Procedia Computer Science*, 80:919–930, 2016.

[128] Sage Management. Second-order Closure Integrated Puff, 2000.

[129] S. Sager. A benchmark library of mixed-integer optimal control problems. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 631–670. Springer New York, 2012.

[130] M. Schwarz, U. Kiencke, T. E. Hodrus, and V. Krebs. Time Discretization of Piecewise Affine Systems with Sliding Modes. *IFAC Proceedings Volumes*, 38(1):302–307, 2005.

[131] G. S. Settles. Fluid Mechanics and Homeland Security. *Annual Review of Fluid Mechanics*, 38:87–110, 2006.

[132] A. Shende, M. J. Bays, and D. J. Stilwell. Toward coordinated sensor motion for classification: An example of intrusion detection using bayes risk. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3340–3346, 2011.

[133] L. Shu, M. Mukherjee, X. Xu, K. Wang, and X. Wu. A Survey on Gas Leakage Source Detection and Boundary Tracking with Wireless Sensor Networks. *IEEE Access*, 4:1–1, 2016.

[134] D. Simon and J. Löfberg. Stability analysis of Model Predictive Controllers using Mixed Integer Linear Programming. In *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, pages 7270–7275, 2016.

[135] V. Šmídl and R. Hofman. Tracking of atmospheric release of pollution using unmanned aerial vehicles. *Atmospheric Environment*, 67:425–436, 2013.

[136] R. N. Smith, J. Das, H. Heidarsson, A. M. Pereira, F. Arrichiello, I. Cetinic, L. Dajarny, M.-È. Garneau, M. D. Howard, C. Oberg, M. Ragan, E. Seubert, E. C. Smith, B. A. Stauffer, A. Schnetzer, G. Toro-Farmer, D. A. Caron, B. H. Jones, and G. S. Sukhatme. USC CINAPS Builds Bridges: Observing and Monitoring the Southern California Bight. *IEEE Robotics and Automation Magazine*, 17(1):20–30, 2010.

[137] R. N. Smith, A. Pereira, Y. Chao, P. P. Li, D. A. Caron, B. H. Jones, and G. S. Sukhatme. Autonomous Underwater Vehicle trajectory design coupled with predictive ocean models: A case study. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4770–4777, may 2010.

[138] Z. Song, Y. Chen, J. Liang, and D. Uciński. Optimal mobile sensor motion planning under nonholonomic constraints for parameter estimation of distributed systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3163–3168, 2005.

[139] J. M. Stockie. The Mathematics of Atmospheric Dispersion Modeling. *SIAM Review*, 53(2):349–372, 2011.

[140] S. Susca, S. Martinez, and F. Bullo. Monitoring environmental boundaries with a robotic sensor network. In *Proceedings of the American Control Conference (ACC)*, 2006.

[141] D. Tolic, R. Fierro, and S. Ferrari. Cooperative Multi-Target Tracking via Hybrid Modeling and Geometric Optimization. In *Proceedings of the 17th Mediterranean Conference on Control and Automation*, pages 440–445, 2009.

[142] F. Treede. *Simulation und Steuerung eines Tragflächenflugzeugs mit ROS/Gazebo*. Bachelor thesis, Technische Universität Darmstadt, 2015.

[143] F. Trespalacios and I. E. Grossmann. Improved Big-M reformulation for generalized disjunctive programs. *Computers and Chemical Engineering*, 76:98–103, 2015.

[144] C. Tricaud. *Optimal Sensing and Actuation Policies for Networked Mobile Agents in a Class of Cyber-Physical Systems*. PhD Thesis, Utah State University, 2010.

[145] C. Tricaud and Y. Chen. Optimal Real-Time Estimation Strategies for a Class of Cyber-Physical Systems Using Networked Mobile Sensors and Actuators. In *Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions*, chapter 10, pages 203–221. 2009.

[146] C. Tricaud and Y. Chen. *Optimal Mobile Sensing and Actuation Policies in Cyber-physical Systems*. Springer, 2011.

[147] C. Tricaud, M. Patan, D. Uciński, and Y. Chen. D-optimal trajectory design of heterogeneous mobile sensors for parameter estimation of distributed systems. In *Proceedings of the American Control Conference (ACC)*, pages 663–668, 2008.

[148] P. Trodden and A. Richards. Adaptive Cooperation in Robust Distributed Model Predictive Control. In *Proceedings of the IEEE International Symposium on Intelligent Control*, number 1, pages 896–901, 2009.

[149] D. Uciński. *Optimal Measurement Methods for Distributed Parameter System Identification*. CRC Press, Boca Raton, 2005.

[150] H. P. Williams. *Model building in mathematical programming*. John Wiley & Sons, Ltd, 5th edition, 2013.

[151] Z. Yan, N. Jouandeau, and A. A. Cherif. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10, 2013.

[152] B. Zhang and G. S. Sukhatme. Adaptive Sampling for Estimating a Scalar Field using a Robotic Boat and a Sensor Network. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, pages 3673–3680, 2007.

[153] D. Zhang, C. Colburn, and T. Bewley. Estimation and adaptive observation of environmental plumes. In *Proceedings of the American Control Conference (ACC)*, pages 4281–4286, 2011.

[154] F. Zhang and N. E. Leonard. Generating contour plots using multiple sensor platforms. In *Proceedings of the IEEE Swarm Intelligence Symposium*, pages 309–316, 2005.

[155] T. Zięba and D. Uciński. Mobile Sensor Routing for Parameter Estimation of Distributed Systems Using the Parallel Tunneling Method. *International Journal of Applied Mathematics and Computer Science*, 18(3):307–318, 2008.

Own Publications

Conference Papers

Juliane Euler and Oskar von Stryk. Decentralized Data-Driven Control of Cooperating Sensor-Carrying UAVs in a Multi-Objective Monitoring Scenario. *IFAC-PapersOnLine*, 50(1):15828–15834, 2017.

Juliane Euler and Oskar von Stryk. Optimized Vehicle-Specific Trajectories for Cooperative Process Estimation by Sensor-Equipped UAVs. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3397–3403, 2017.

Tobias Ritter, Juliane Euler, Stefan Ulbrich, and Oskar von Stryk. Decentralized Dynamic Data-Driven Monitoring of Atmospheric Dispersion Processes. *Procedia Computer Science*, 80:919–930, 2016.

Juliane Euler, Tobias Ritter, Stefan Ulbrich, and Oskar von Stryk. Centralized Ensemble-Based Trajectory Planning of Cooperating Sensors for Estimating Atmospheric Dispersion Processes. In Sai Ravela and Adrian Sandu, editors, *Dynamic Data-Driven Environmental Systems Science*, volume 8964 of *Lecture Notes in Computer Science*. Springer International Publishing, 1 edition, 2015.

Tobias Ritter, Juliane Euler, Stefan Ulbrich, and Oskar von Stryk. Adaptive Observation Strategy for Dispersion Process Estimation Using Cooperating Mobile Sensors. *IFAC Proceedings Volumes*, 47(3):5302–5308, 2014.

Juliane Kuhn, Christian Reinl, and Oskar von Stryk. Predictive Control for Multi-Robot Observation of Multiple Moving Targets Based on Discrete-Continuous Linear Models. *IFAC Proceedings Volumes*, 44(1):257–262, 2011.

Workshop Papers

Juliane Euler, Andreas Horn, Dominik Haumann, Jürgen Adamy, and Oskar von Stryk. Cooperative N-Boundary Tracking in Large Scale Environments. In *Proceedings of the IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems*, 2012.

Wissenschaftlicher Werdegang[1]

06/2004	Allgemeine Hochschulreife
10/2004 – 12/2009	Studium der Mathematik an der Technischen Universität Darmstadt
12/2009	Diplom der Mathematik
seit 02/2010	Wissenschaftliche Mitarbeiterin und Doktorandin, Fachbereich Informatik, Technische Universität Darmstadt

Erklärung[2]

Hiermit erkläre ich, dass ich die vorliegende Arbeit, mit Ausnahme der ausdrücklich genannten Hilfsmittel, selbständig verfasst habe.

[1] gemäß § 20 Abs. 3 der Promotionsordnung der TU Darmstadt
[2] gemäß § 9 Abs. 1 der Promotionsordnung der TU Darmstadt